

UNIVERSITETET I OSLO
Fysisk institutt

**Estimering av
MEMS-
gyroparametre**

Masteroppgave

Jørn Skarbø Grahn

30. mai 2011



Forord

Denne oppgaven er et resultat av den avsluttende delen av min masterutdanning, ved Universitetet i Oslo. Arbeidet har i hovedsak blitt utført på UNIK, Universitetssenteret på Kjeller. Oppgaven er utført i samarbeid med FFI, hvor Ørnulf Kandola og Anders Rødningsby, har vært mine eksterne veiledere. Jeg vil rette en takk til disse to for god hjelp og utlån av gyroskop.

Spesielt vil jeg takke min internveileder, Oddvar Hallingstad, for god støtte og faglig oppfølging underveis i arbeidet. All hjelpen har ført til at arbeidet har vært ekstra lærerikt.

Kjeller, 30.05.2011

Jørn Skarbø Grahn

Sammendrag

I denne oppgaven er det estimert forskjellige gyroparametre. Parametrene som er estimert, sier noe om ytelsen og nøyaktigheten til et tilhørende gyroskop. Det er tatt utgangspunkt i gyroskopene SAR100 og STIM202 da disse er lånt fra FFI. Det er blitt utført målinger på de to gyroskopene slik at kunnskaper om signalene er opparbeidet. På bakgrunn av denne kunnskapen er det utviklet matematiske modeller for forskjellige typer av gyrosignal. For at de simulerte signalene skal være så realistiske som mulig, er det gjort tilpasninger av disse, slik at det er stor visuell likhet mellom simulert og reelt gyrosignal. Det antas derfor at de matematiske modellene beskriver de reelle signalene med stor nøyaktighet.

Estimeringsmetodene som er brukt i oppgaven er allan variansen, og maximum likelihood estimering. I allan variansen er parametre blitt estimert i såkalte allan varians plot. Plotene fremkommer ved hjelp av målinger fra gyroskopene. Flere av disse estimatene stemte overrens med verdier oppgitt i de tilhørende databladene.

Den matematiske modellen er blitt beskrevet på en kontinuerlig form slik at den kunne implementeres i et kalmanfilter. I filteret er det også implementert likninger som muliggjør maximum likelihood estimeringen. Det er gjort flere estimater ved bruk av måleserier fra gyroskopene. Det er påvist konvergens i flere av de resulterende estimatene fra maximum likelihood metoden. Det er også plotet autokorrelasjon av de forskjellige gyrosignalene. Informasjonen som fremkommer av disse plotene, er med på å identifisere tilstedeværelsen av forskjellige parametre i signalene.

Innhold

1	Innledning	1
1.1	MEMS-sensorer	1
2	MEMS-Gyroskop	3
2.1	Corioliskraften	3
2.2	Utforminger av MEMS-Gyroskop	4
2.2.1	Enkel Oscillator	6
2.2.2	Balansert oscillator	6
2.2.3	Ring oscillator	9
2.2.4	Begrensninger	10
2.3	Målemetoder	11
2.3.1	STIM202	11
2.3.2	SAR100	12
2.3.3	Oppsummering	13
3	Identifikasjon	15
3.1	Støy	15
3.1.1	Hvit støy	15
3.1.2	Farget støy	16
3.2	Gyroparametre	17
3.2.1	Angular Random Walk (ARW)	17
3.2.2	Rate Random Walk (RRW)	19
3.2.3	Bias instability	19
3.3	Simuleringsmodell som genererer målinger	19
3.3.1	Diskretisering	21
3.3.2	Simulering av gyroskopet STIM202	23
3.4	Simulering av gyroskopet SAR100	24
3.5	Allan variansen	25
3.5.1	Oppsummering	27
3.6	Maximum Likelihood estimering	28
3.6.1	Prinsipp	28

3.7	Kalmanfilter	29
3.7.1	Kontinuerlig Kalmanfilter	30
3.7.2	Diskret Kalmanfilter	30
3.7.3	Kontinuerlig diskret Kalmanfilter	31
3.8	Stokastiske variabler	31
3.9	Sannsynlighetsfordeling	32
3.10	MLE anvendt på gyromålinger	33
3.10.1	Implementering av parametervektoren	36
3.10.2	Test av MLE på simulert gyrodata	36
3.11	Autokorrelasjon	40
4	Resultater	45
4.1	Allanvarians	45
4.1.1	SAR100	45
4.1.2	STIM202	46
4.2	Maximum likelihood	47
4.2.1	SAR100	48
4.2.2	STIM202	49
4.3	Autokorrelasjon	55
4.4	Diskusjon	57
5	Konklusjon	59
6	Videre arbeid	61
A	Pseudokode	65
A.1	Simulering	65
A.1.1	Kall av funksjon	66
A.1.2	Minimalisering	67
B	Datablad	69
C	MATLAB-kode	75
C.1	Simuleringsprogram	75
C.2	Allanvarians	77
C.3	Maximum likelihood metoden	95
C.4	Autokorrelasjon	97

Figurer

2.1	Generering av corioliskraften.	5
2.2	Forskjellige utforminger av MEMS-oscillator [11].	6
2.3	Første fungerende prototype av et stemmegaffel gyroskop, laget av The Charles Stark Draper Laboratory Inc. [11].	7
2.4	Bevegelsen til masse-elementene i et stemmegaffel-gyroskop [11].	7
2.5	Komponenter i en balansert oscillator (stemmegaffel-gyroskop) [11].	8
2.6	Blokk diagram til stemmegaffel-gyroskopet.	9
2.7	Snitt av gyroskop med vibrerende silikonring [11].	9
2.8	Gyroskop med vibrerende silikonring [11].	10
2.9	Tre-akset STIM202 med akse-orientering.	12
2.10	SAR100 innkapslet i LCC pakke, samt den innvendige elementstrukturen.	13
2.11	SAR100 blokkdiagram.	13
3.1	Eksempel på en gaussisk hvitstøy prosess.	16
3.2	Log-log graf for allan variansen ved ARW [9].	18
3.3	Eksempel på en RRW prosess.	19
3.4	Simulert ARW prosess, RRW prosess, farget støy (1.ordens markov prosess) og resulterende simulert gyrosignal.	23
3.5	Simulert gyrosignal med redusert bidrag fra ARW prosess, samt mindre korrelasjonstid ,T.	24
3.6	Simulert og målt signal.	25
3.7	Målt gyrosignal fra SAR100 med og uten bias.	26
3.8	Log-log graf for allanvariansen med alle feilkilder [9].	27
3.9	Normalfordelingen.	34
3.10	Autokorrelasjon simulert gyrosignal.	42
3.11	Et nærmere bilde av autokorrelasjonen for simulert gyrosignal.	43
3.12	Et nærmere bilde av autokorrelasjonen for hvitstøy.	43
4.1	Allanvarians plot SAR100.	46

4.2	Allanvarians plot for z-aksen.	47
4.3	Allanvarians plot for x-aksen.	47
4.4	Allanvarians plot for y-aksen.	48
4.5	Plot av konvergens (funksjonsvurdering) for parametrene til SAR100.	48
4.6	Plot av konvergens (funksjonsvurdering) for parametrene til SAR100, med nye verdier av p_0	49
4.7	Plot av konvergens (funksjonsvurdering) for parametrene for z-aksen	50
4.8	Plot av konvergens (funksjonsvurdering) for parametrene for z-aksen, med nye verdier av p_0	51
4.9	Plot av konvergens (funksjonsvurdering) for parametrene for x-aksen	52
4.10	Plot av konvergens (funksjonsvurdering) for parametrene for x-aksen, med nye verdier av (p_0)	53
4.11	Plot av konvergens (funksjonsvurdering) for parametrene for y-aksen.	53
4.12	Plot av konvergens (funksjonsvurdering) for parametrene for y-aksen, med nye verdier av (p_0)	54
4.13	Autokorrelasjon for SAR100 og STIM202.	55
4.14	Et nærmere bilde av autokorrelasjonen til STIM202.	55
4.15	Autokorrelasjon av signal fra SAR100 uten bias.	56

Tabeller

2.1	Nøkkelparametre for gyroskopene.	13
3.1	Resultater av forsøk 1.	37
3.2	Resultater av forsøk 2.	38
3.3	Resultater av forsøk 3.	38
3.4	Resultater av forsøk 4.	39
3.5	Resultater av forsøk 5.	39
3.6	Resultater av forsøk 6.	40
4.1	Resultater fra allan varians for SAR100.	45
4.2	Parametre for hver av aksene.	46
4.3	Estimerte verdier for SAR100	49
4.4	Estimerte verdier til SAR100, med nye verdier av p_0	50
4.5	Estimerte verdier til z-aksen	50
4.6	Estimerte verdier til z-aksen, med nye verdier av (p_0)	51
4.7	Estimerte verdier til x-aksen	51
4.8	Estimerte verdier til x-aksen, med nye verdier av (p_0)	52
4.9	Estimerte verdier til y-aksen.	52
4.10	Estimerte verdier til y-aksen, med nye verdier av (p_0)	54
4.11	Oppsummering av estimerte verdier.	57

Kapittel 1

Innledning

MEMS er en forkortelse for Mikro Elektro-Mekaniske Systemer (Micro-machined electromechanical-systems) [11]. MEMS-systemer kan bli betraktet som små mekaniske systemer, som er drevet av elektrisitet. Disse unike egenskapene fører til at slike systemer både har elektriske og mekaniske funksjoner. Komponentene som inngår varierer mellom 1-100 μm i størrelse. MEMS-enheter som helhet, er derfor av en typisk størrelsesorden fra 20 μm til 1 mm.

1.1 MEMS-sensorer

Innføring av applikasjoner som krever lav-kost sensorer for måling av akselerasjon og vinkelfart, har resultert i en ansporende utvikling av MEMS-sensorer. Dagens MEMS-enheter overvinner mange funksjonelle hindringer som før var forbundet ved å ta i bruk disse, i forskjellige applikasjoner. Dette gjelder særlig de applikasjonene der pris, størrelse, og strømforbruk spiller en stor rolle.

MEMS-sensorer produseres i store volum. Dette medfører at de er billige å produsere, samtidig som kravene til nøyaktighet blir opprettholdt. Den lille størrelsen av enhetene fører til en lav vekt, og et lavt strømforbruk.

Produksjonen av de konvensjonelle gyroskopene har blitt automatisert med stor suksess. Prisen på slike gyroskop er fortsatt høy på grunn av de store kostnadene forbundet med montering, og testing av de innvendige komponentene. Faktorer som dette har resultert i en stor spredning av ulike applikasjoner, både militære og kommersielle, hvor MEMS-enheter kan brukes.

Formålet med denne oppgaven er å estimere gyroparametre. For at dette skal kunne realiseres, er det nødvendig å ha grunnleggende kunnskap om

støybidrag i signaler fra gyroskop. Det kan også tenkes at det kan lages en matematisk modell til gyroskopet, på bakgrunn av kunnskapen som er opparbeidet. Estimeringsmetodene som skal anvendes er allan variansen og maximum likelihood metoden. Det vil også bli regnet ut autokorrelasjon til signalene, slik at støybidrag kan verifiseres.

Kapittel 2

MEMS-Gyroskop

Dette kapittelet er ment som en generell teoretisk innføring i teori om MEMS-gyroskop. Det er tatt utgangspunkt i tre forskjellige utforminger av denne typen gyroskop. Her er det virkemåte av disse enhetene som er det essensielle. Det er ikke forutsatt at leseren skal kjenne til dette, men oppgaven kan bli forstått i et bedre perspektiv.

2.1 Corioliskraften

Gyroskop generelt, er enheter som måler vinkelfart. MEMS-gyroskop er ikke-roterende enheter som benytter seg av coriolis kraften for å gjøre slike målinger [2].

Coriolis effekten er en kraft som i fysikken kan innføres for å få et system på jorden til å oppføre seg som et treghetssystem. Når jorden dreier, vil et punkt på ekvator (magnetisk ekvator) bevege seg mye hurtigere enn et punkt ved en av polene. Denne farten vil øke fra tilnærmet null ved polene til nesten 1700 km i timen ved ekvator. På bakgrunn av denne bestemte rotasjonen vil corioliseffekten oppstå. Vind og havstrømmer vil naturlig nok bli påvirket av denne kraften slik at det skapes helt bestemte mønstre på jorden.

Effekten vil føre til en tilsynelatende nedbøying av bevegelige objekter når de blir sett fra et roterende referansesystem. Coriolis kraften er forårsaket av jordrotasjonen og tregheten i en masse som opplever effekten. Kraften er proporsjonal med objektets masse og rotasjonsraten. Formelen for styrke og retning av coriolis akselerasjon, på vektorform kan sees under

$$\mathbf{a}_c = -2\boldsymbol{\Omega} \times \mathbf{v} \quad (2.1)$$

hvor a_c er partikkelens akselerasjon, og v er partikkelens fart i den roterende rammen. Ω er vektoren for vinkelfart, og den har lik størrelse som

rotasjonsraten ω . Vektoren er rettet langs rotasjonsaksen i det roterende referansesystemet.

Ved å multiplisere 2.1 med massen m til det relevante objektet, kan coriolis kraften uttrykkes som

$$\mathbf{F}_c = -2m\boldsymbol{\Omega} \times \mathbf{v} \quad (2.2)$$

Kryssproduktet mellom vektorene kan bli betraktet som determinanten til en matrise:

$$\begin{bmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ \Omega_x & \Omega_y & \Omega_z \\ v_x & v_y & v_z \end{bmatrix} = \begin{pmatrix} \Omega_y v_z & -\Omega_z v_y \\ \Omega_z v_x & -\Omega_x v_z \\ \Omega_x v_y & -\Omega_y v_x \end{pmatrix} \quad (2.3)$$

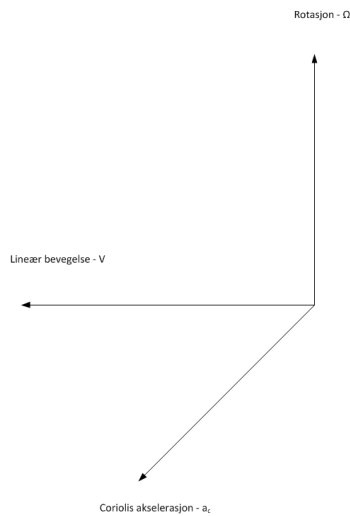
hvor i, j, k er enhetsvektorene i x, y og z-retning.

Coriolis akselerasjonen vil både virke vinkelrett på rotasjonsaksen, og på objektets hastighet i den roterende rammen. Dette vil gi forskjellige retninger på akselerasjonen avhengig av objektets hastighet.

I MEMS-gyroskop brukes denne kraften til å oppfatte dreining. I likhet med objektene her på jorden, vil de små elementene i gyroskopene bli påvirket av kraften som oppstår på grunn av en generert rotasjon av enheten. Det finnes forskjellige utforminger og varianter av MEMS-gyroskop. I de neste avsnittene vil det bli forklart tre utforminger som utnytter denne kraften.

2.2 Utforminger av MEMS-Gyroskop

Mye av utviklingen som har blitt gjort angående ytelsen til gyroskopene kan tilskrives The Charles Stark Draper Laboratories Inc, i USA [11]. Over de siste 50 årene har de funnet opp, og utviklet TNS, som brukes på jorda og i verdensrommet. Laboratoriet har også utviklet MEMS gyroskop de siste 10 årene, og de var også først ute med å demonstrere måling av en rotasjons-rate ved bruk av en mikro-maskinert silikon sensor. I 1992 lyktes deres ingeniører i å implementere et coriolis-vibrerende gyroskop ved bruk av MEMS teknologi. Siden den gang har det foregått en kontinuerlig utviklingsprosess innen slike enheter. Dette har blant annet resultert i produksjon av gyroskop med bias-stabiliteter i området 0.2-10 °/h. Mye av denne utviklingen har hatt sitt siktemål mot militære applikasjoner, noe som fortsatt er drivkraften bak den stadige utviklingen av ytelsen til enhetene. Kommersiell ytelse og bruk, har også blitt relevant de seneste årene. Teknologien tilpasser seg derfor også lavkost produksjon av store volum, av enheter med mer beskjeden ytelse. I

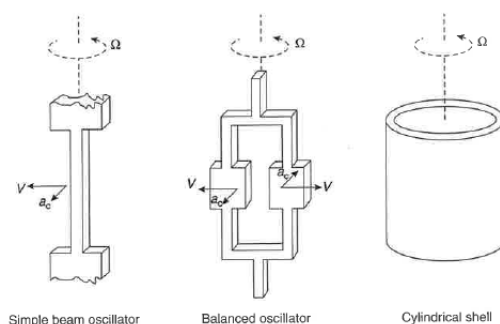


Figur 2.1: Generering av corioliskraften.

tiden fremover er bilindustrien ansett for å være en av de største brukerne av slike sensorer. Grunnen til dette, er den stadige implementeringen av slike sensorer i biler. Idag finnes det gyroskop i blant annet anti-skrens applikasjoner, abs-systemer, og navigasjonssystemer.

Som sagt tidligere, benytter MEMS-gyroskop seg av coriolis kraften, for å oppfatte rotasjon. Dette gjøres ved at en eller flere faste vibrerende masser, blir utsatt for en coriolis kraft, slik at rotasjon kan måles. De to massene vil da rotere i planet med en frekvens ω_r . Coriolis kraften som massene blir utsatt for, vil forårsake en akselerasjon på de to massene som tilsvarer likning 2.1. Sensorene vil da oppdage kraften som virker på den vibrerende massen, gitt som en lineær vibrerende bevegelse i en referanseramme. Referanserammen roterer samtidig rundt en akse. Denne aksen står vinkelrett på aksen til den lineære vibrerende bevegelsen. I figur 2.1 er det illustrert en generert coriolis kraft. Siden MEMS-gyroskop benytter seg av coriolis kraften, er det viktig å se hvilken presisjon man kan forvente i dens tilhørende målinger. Eksempelvis vil en typisk MEMS gyro med input på 1 rad/s, resultere i en corioliskraft på ca 9×10^{-8} N. Dette vil gi et toppunkt i bevegelsen langs den følsomme aksen på 10^{-9} m, noe som igjen vil føre til en forandring av kapasitansen på 3×10^{-18} F.

Det finnes mange praktiske konfigurasjoner av sensorer som er basert på prinsippet beskrevet over. En kan vanligvis dele slike sensorer inn i tre kategorier:



Figur 2.2: Forskjellige utforminger av MEMS-oscillator [11].

- Enkel oscillator
- Balansert oscillator (ofte kalt stemmegaffel-gyroskop)
- Ring oscillator (ofte kalt sylinder eller vin glass gyroskop)

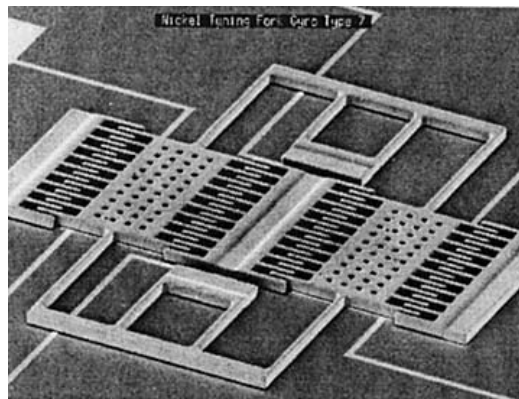
2.2.1 Enkel Oscillator

En enkel oscillator utgjør den mest simple typen av vibrerende gyroskop. Slike gyroskop baserer seg på en oscillator, som alene, kan bli modellert som en enkelt vibrerende masse. Ulempene i en slik modell oppstår gjennom mekanisk asymmetri i konstruksjonen av oscillatoren. Dette vil blant annet resultere i en ekstra kopling mellom aksene som vibrerer og måleaksen. Det er også følsomhet for eksterne vibrasjoner, noe som vil gi en gjensidig påvirkning mellom det vibrerende elementet og strukturen det er påmontert. Hvis etuiet til sensoren blir påvirket av støt eller vibrasjoner vil dette også bli plukket opp av det vibrerende elementet.

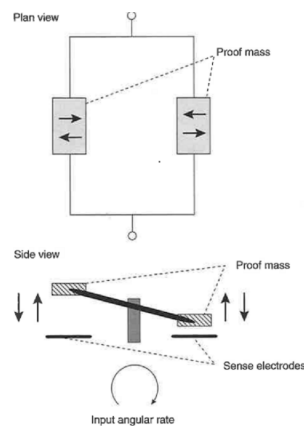
2.2.2 Balansert oscillator

Problemene forbundet med sensitivitet ovenfor eksterne vibrasjoner i en enkel oscillator, kan unngås ved å ta i bruk en balansert oscillator. Slike enheter er bygget opp av enten quartz eller silikon. Bruk av denne typen materiale, har ført til en signifikant forbedring av ytelse de siste årene.

Stemmegaffel gyroskopet i figur 2.3, var bestående av en silikon struktur montert på et underlag av glass. Silikon strukturen inneholder to masser, hengt opp av en sekvens av bjelker, som igjen er forankret til underlaget ved



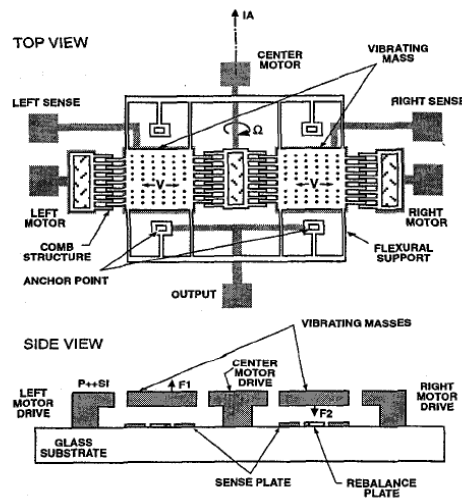
Figur 2.3: Første fungerende prototype av et stemmegaffel gyroskop, laget av The Charles Stark Draper Laboratory Inc. [11].



Figur 2.4: Bevegelsen til masse-elementene i et stemmegaffel-gyroskop [11].

bestemte punkter. De to massene er laget på en slik måte at de oscillerer 180 grader ut av fase, når spenning blir tilført til noen ytre motorer. De vil oscillere med lik amplitude, men i motsatt retning i forhold til hverandre. Denne funksjonen har gitt sensoren betegnelsen ”stemmegaffel gyroskop”. En skisse av bevegelsene til de to massene kan sees i figur 2.4 .

Hvis en genererer en vinkelfart om input-aksen, vil dette gi opphav til en coriolis-kraft som vil presse massene inn og ut av planet de oscillerer i. De påfølgende hastighetsvektorene til massene er like, men motsatt rettet. Dette vil føre til en ikke-parallell bevegelse forårsaket av coriolis-kraften. Denne bevegelsen måles ved hjelp av kondensator plater som er plassert over og under hver av de to massene. Forskyvningen av massene vekk fra planet, vil deretter bli målt, slik at det fremkommer et signal knyttet til systemets rotasjon. Signalet er proporsjonalt med vinkelfarten som blir anvendt rundt input-aksen.



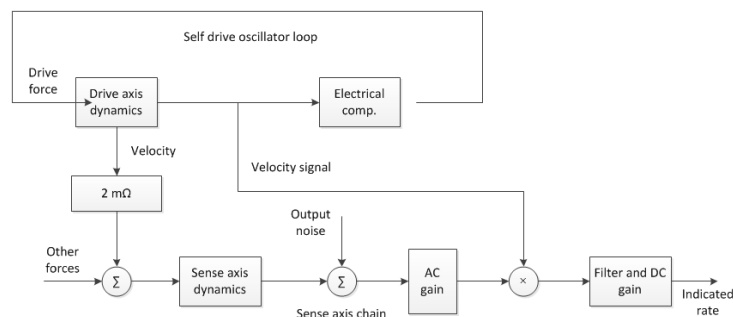
Figur 2.5: Komponenter i en balansert oscillator (stemmegaffel-gyroskop) [11].

En skjematisk fremstilling av komponentene som inngår i stemmegaffel gyroskopet er vist i figur 2.5.

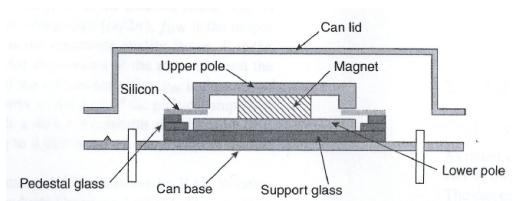
Størrelsen på de faste masse-elementene som ble brukt i dette gyroskopet var $1000 \mu\text{m} \times 1000 \mu\text{m} \times 20 \mu\text{m}$. Disse kan sees i figur 2.5 merket som F1 og F2. Referansefrekvensen til oscillasjonen av massene er på 12 kHz, og amplituden 10 μm . Når enheten roteres vil massene vibrere ortogonalt ut av planet, og frekvensen og amplituden vil følgelig få en høyere verdi.

For alle gyroskop er designen av elektronikken som brukes til å drive og kontrollere sensorene, et kritisk sub-system. Det kreves at komponenter som inngår i disse systemene er optimalisert på en slik måte at de vil gi et stabilt og repeterbart ut-signal. Følgelig vil også den tilhørende ytelsen til gyroskopet være god og pålitelig.

Sub-systemet til gyroskopet som ble beskrevet ovenfor, utgjøres av en motor sløyfe og sensorer. Motor sløyfen omfatter en selvdrevet oscillator, som tilføres spenning slik at det blir generert elektrostatisk kraft, som igjen induserer bevegelser (sidelengs) til hver av massene. Denne bevegelsen blir deretter oppfattet og kontrollert av sensorene slik at det opprettholdes en konstant amplitude til bevegelsen. Massene anses da for å være i null-posisjon. Påvisning av den induserte coriolis-bevegelsen blir gjennomført ved bruk av en svært sensitiv kapasitans-sensor. En forsterker oppfatter den påfølgende belastningen som genereres ved forflytningen av masse-elementene. Utdata-en fra sub-systemet er et signal som er proporsjonalt med tilførselen. Et



Figur 2.6: Blokk diagram til stemmegaffel-gyroskopet.



Figur 2.7: Snitt av gyroskop med vibrerende silikonring [11].

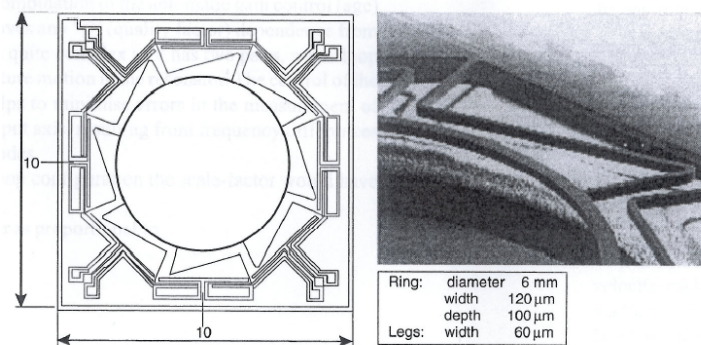
blokkskjema av elektronikken som inngår er vist i figur 2.6

2.2.3 Ring oscillator

Den siste utformingen av MEMS-gyroskopene som skal forklares er en ring oscillator. I motsetning til de enkle oscillatorene er dette gyroskopet symmetrisk om den roterende aksen. Enheter som baserer seg på en vibrerende ring eller sylinder, utgjør denne kategorien. De er relativt enkle å produsere til en høy grad av nøyaktighet.

Denne typen av MEMS gyroskop detekterer dreining om en akse, som står vinkelrett på planet til den tilhørende ringen i gyroskopet. Deteksjon av dreiningen blir gjort ved hjelp av en kopling til dette planet, slik at forskyvninger blir oppdaget i form av coriolis-kraft. En slik sensor har en fordel ved at strukturen til ringen er konstruert på en slik måte at vibreringen av ringen opprettholdes, og en kan på denne måten måle vibrasjonsenergien i det samme planet. Ulempen derimot, er den lave frekvensen til vibrasjonen av massen, som vil føre til en lav skala faktor.

Gyroskopet, som er vist i figur 2.7 og 2.8, er laget av silikon. Det skapes



Figur 2.8: Gyroskop med vibrerende silikonring [11].

en elektrostatisk vibrering av ringen ved at strøm blir tilført til metallsporene som befinner seg ovenpå ringen, og at et magnetfelt blir stående vinkelrett på sensorplanet. Diameteren til ringen er 6 mm, og den grunnleggende vibrasjonen har en frekvens 14,5 kHz. Det følgende elementet har 8 identiske elektrisk ledende sløyfer, som følger identiske baner på overflaten av silikonet. Sløyfene går fra et lednings-ben, langs lengden av et av støttebenene til ringen, deretter rundt en åttendedel av silikon ringen, så ned igjen langs lengden til det neste støttebenet, og til slutt tilbake til lednings-benet. Det går også egne ledere mellom støttebenene slik at det unngås kapasitans-koplinger mellom dem. Jordplanet er opprettet gjennom en kopling til silikonskroget. Når det blir tilført strøm til lederbenene vil ringen oppnå resonans. Den påfølgende Coriolis-induserte bevegelsen til ringen blir oppdaget av de induserte spenningene som oppstår, ettersom benene ”kutter” det magnetiske feltet. En illustrasjon av denne strukturen er vist i figur 2.8. Lednings-benene måles for å oppfatte rotasjonen som utføres på systemet.

Den vibrerende ringen er designet slik at all bevegelse forbundet med vibreringen finner sted i planet. Denne egenskapen er med på å gjøre denne typen gyroskop ufølsomme mot omgivende bevegelser og støy av normal båndbredde. Dette er svært praktisk da slike bevegelser inngår i mange av bruksområdene til gyroskopet.

2.2.4 Begrensninger

Som sagt har MEMS-gyroskop blitt utviklet mye de siste 10 årene, særlig etter de har blitt tatt i bruk kommersielt. Dette har ført til en utvikling av svært små enheter. De nåværende gyroskopene består ofte av kretskort med tykkelse helt ned til 15 μm av forskjellig materiale som blant annet krystallsilikon og polysilikon [10]. En slik tykkelse begrenser frekvensen til

referansebevegelsen til 10kHz. I og med at hørevidden til mennesker er mellom 10-20kHz betyr dette at gyroskopene vil være sensitivt ovenfor akustisk støy og vibrasjoner fra omgivelsene. En mobiltelefon vil for eksempel kunne bli påvirket av ringetoner og vibrasjoner. Andre begrensninger kan være reduksjoner etter fabrikasjoner, og miljøvariasjoner, som fører til en variasjon mellom oscillasjonsfrekvensen til massene..

2.3 Målemetoder

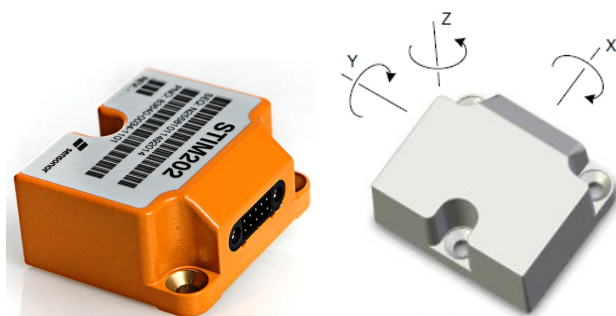
I denne oppgaven er det tatt utgangspunkt i to forskjellige gyroskop, SAR100 og STIM202. Gyroskopene er lånt fra FFI og de er av forskjellig nøyaktighet og størrelse. Det ene gyroskopet er en-akset (SAR100) og det andre er tre-akset (STIM202). Måling på de to gyroskopene ble gjennomført på lik måte. Gyroskopet ble plassert på et bord, stående i ro. Det antas at bordet er parallelt med horisontalplanet slik at også akse som står vinkelrett på dette planet er parallell med tyngdekraftvektoren. Siden STIM202 har tre akser, ble gyroskopet satt i forskjellige posisjoner ved hver måling, slik at hver akse som det ble gjort målinger på, var parallell med tyngdekraftvektoren. Dette ble gjort slik at hver av aksene ble utsatt for like krefter under måling. Siden parametre for hver av aksene skal estimeres utifra rådata, er det også en fordel at målingene fremkommer på samme måte for hver akse. Resultatene er på denne måten direkte sammenlignbare.

2.3.1 STIM202

Gyroskopet STIM202 er en "liten pakke" bestående av tre svært nøyaktige MEMS-gyroskop. Hver av aksene er fabrikk-kalibrert med hensyn på bias feil og følsomhet, og det er i tillegg kompensert for eventuelle temperatureffekter slik at det er høy nøyaktighet i målingene. Temperaturspennet for slike målinger er i området -40°C til $+85^{\circ}\text{C}$.

Inndata for gyroskopet er $\pm 400^{\circ}/\text{s}$, og utdata kan settes til å være vinkel-fart eller inkrementell vinkel. Gyroskopet er tunet med hensyn på eksitasjon og deteksjon frekvenser, og det har perfekte balanserte vibrasjons masser. Dette fører til lav sensitivitet når det gjelder vibrasjoner og støt fra omgivelsene. Gyroskopet blir drevet av en strømforsyning på $+5\text{V}$ som blir tilført gjennom en USB-kabel som er tilkoblet en PC. Den kommuniserer gjennom et RS422 grensesnitt hvor det blir overført måledata. Når gyroskopet blir tilkoblet vil det gjennomføre en intern systemsjekk og synkronisere gyroene.

Måledataene overføres i form av pakker med et fast format, og med et



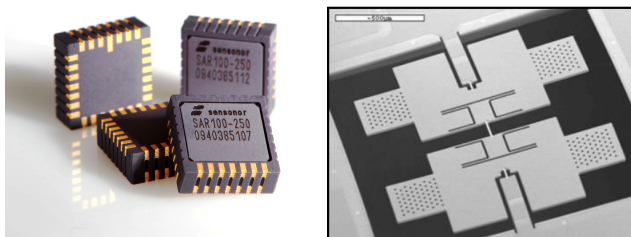
Figur 2.9: Tre-akset STIM202 med akse-orientering.

intervall gitt av samplingstiden. Disse datagrammene er i binærformat, da dette skal effektivisere overføringen. I tillegg til måledata inneholder disse datagrammene blant annet en statusbyte som skal detektere feil i overføringene eller systemet. Det er også mulig å starte gyroskopet i noe som heter ”service mode”. Her kan alle konfigurasjons parametrene bli midlertidig eller permanent forandret ved at de blir overskrevet i gyroskopets flash memory. Samplingstiden er et eksempel på en slik parameter.

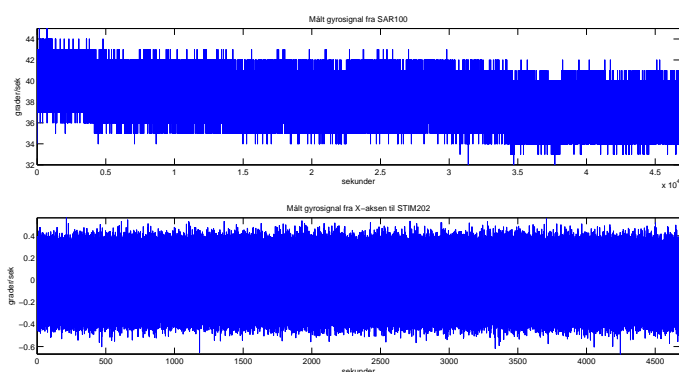
2.3.2 SAR100

Gyroskopet SAR100 er en en-akset digital sensor som er innkapslet i en keramisk LCC pakke slik at den kan monteres både vertikalt og horisontalt. Gyroskopet er fabrikk-kalibrert og kompensert for temperatureffekter slik at målingene er nøyaktige over et bredt temperaturspekter. Inndata for sensoren er ± 100 °/s. Kommunikasjon mellom gyroskopet og applikasjoner skjer gjennom et SPI grensesnitt.

Funksjonen for signalet fra gyroskopet er basert på eksitasjon av en referansebevegelse som finner sted i den ”sommerfugl” lignende konstruksjonen. En illustrasjon av dette kan sees i 2.10. Når gyroskopet roteres vil det bli generert coriolis-krefter, som har samme frekvens som referanse bevegelsen til elementene. Rotasjonen vil derfor være et mål av amplituden til den resulterende vibrerende bevegelsen av massene. Gyroskopet er lite sensitivt mot vibrasjoner og støt fra omgivelsene. Grunnen til dette er den gode mekaniske og elektriske balanseringen av massene, samt tuning av eksitasjon og deteksjon frekvenser.



Figur 2.10: SAR100 innkapslet i LCC pakke, samt den innvendige elementstrukturen.



Figur 2.11: SAR100 blokkdiagram.

2.3.3 Oppsummering

I tabell 2.1 er det gjort en sammenlignende oversikt av nøkkel-parametre fra databladene til de to gyroskopene. Det er tydelig at STIM202 er mer nøyaktig enn SAR100. Et utdrag av databladene kan sees i vedlegg B. I figur ?? er det plotet to målte signaler, hvor det ene er fra SAR100, og det andre fra x-aksen til STIM202. I de neste kapitlene skal blant annet disse parametrene estimeres med forskjellige metoder.

Tabell 2.1: Nøkkelparametre for gyroskopene.

	Angular Random Walk	Bias Instability
STIM202	$0.2 \text{ } ^\circ/\sqrt{h}$	$0.5 \text{ } ^\circ/h$
SAR100	$0.65 \text{ } ^\circ/\sqrt{h}$	$50 \text{ } ^\circ/h$

Kapittel 3

Identifikasjon

I dette kapitlet vil det bli forklart tre forskjellige identifikasjonsmetoder. Alle metodene inngår som en del av den totale parameterestimeringen til gyroparametrene. De forskjellige gyroparametrene som skal estimeres, samt den tilhørende simuleringsmodellen, vil bli forklart i detalj i kapitlet.

3.1 Støy

Det vil her bli gjennomgått to forskjellige støytyper som er brukt i simuleringen av gyrosignaler.

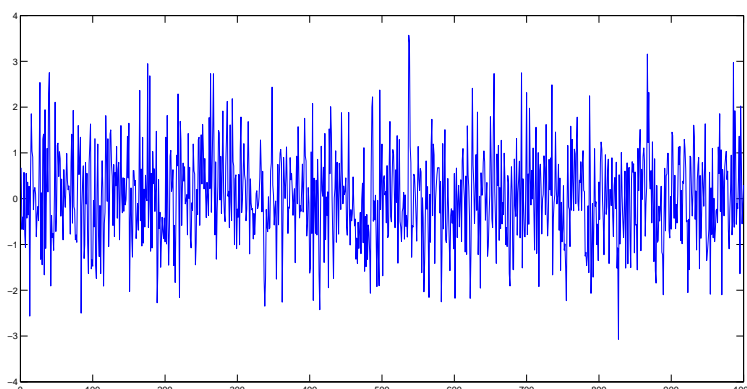
3.1.1 Hvit støy

Hvit støy er definert som et stokastisk signal. Det vil si at hvis en kjenner signalet frem til et gitt tidspunkt, vil ikke dette gi noen indikasjon på hvordan signalet vil se ut i neste tidspunkt. En kan på denne måten si at hvit støy er karakterisert ved å være tilfeldig, og ha null som gjennomsnittsverdi. Denne støyen skilles fra andre støytyper ved at alle frekvenser er likt representert i hvitstøy.

Hvit støy brukes ofte når en skal modellere støy fra ulike sensorer, som for eksempel gyroskop. Det er også vanlig at denne støytypen inngår når andre støytyper skal modelleres. Autokorrelasjonen for hvitstøy er definert som

$$R_{\omega}(t, s) = Q(t)\delta(t - s) \quad (3.1)$$

hvor $Q(t)$ er en vilkårlig skaleringskonstant og $\delta(t)$ er dirac delta funksjonen som har verdi 0 for alle t unntatt $t=0$ [12]. Integralet av denne funksjonen $= 1$. Gaussisk hvit støy forekommer når støyen er karakterisert ved en



Figur 3.1: Eksempel på en gaussisk hvitstøy prosess.

gaussisk fordeling. Dette er illustrert i figur 3.1 I det stasjonære tilfellet vil autokorrelasjonen ha denne formen

$$R_{\omega}(\tau) = Q\delta(\tau) \quad (3.2)$$

Utifra dette kan det vises at hvit støy er like fremtredende i hele frekvensspekteret, og det kan dermed sies at denne støyen har uendelig båndbredde. Dette er bare mulig å konstruere rent teoretisk da båndbredden til hvitstøy er begrenset i praksis.

I det diskrete tilfellet er hvitstøy definert som en hvit sekvens, med ukorrelerte tilfeldige verdier, og middelværdi lik null. Autokorrelasjonen i det diskrete tilfellet uttrykkes på formen

$$E[\omega_m \omega_l] = Q\delta_{ml} \quad (3.3)$$

hvor $\delta_{ml} = 1$ hvis $m=l$ og 0 ellers.

3.1.2 Farget støy

Den andre støytypen som skal beskrives er farget støy. Ved å filtrere hvitstøy er det mulig å generere en spesiell type av støyprosesser, kalt Gauss-Markov prosess. Prosessen kan genereres av forskjellig orden utifra hvilken orden det er på filtret som brukes.

En kontinuerlig prosess $x(t)$ er en 1. ordens markov for alle k og $t_1 < t_2 < \dots < t_k$ når

$$F[x(t_k)|x(t_{k-1}), \dots, x(t_1)] = F[x(t_k)|x(t_{k-1})]$$

Uttrykket over viser at sannsynlighetsfordelingen for prosessen $x(t)$, bare er avhengig av verdien et tidsskritt tilbake i tid, $x(t_{k-1})$ [5]. Dette betyr at det kun er sannsynlighetsfordelingen til nåværende tilstand som har betydning for den fremtidige tilstanden. Siden denne prosessen er karakterisert som en 1. ordens markov, kan den beskrives med differensiallikningen,

$$\frac{dx}{dt} = -\frac{1}{T}x + w \quad (3.4)$$

hvor $-\frac{1}{T}$ er korrelasjonstiden og w er hvit støy [ref gelb]. Hvis en også antar at sannsynlighetstetthetsfunksjonene til w og x er gaussiske, kan $x(t)$ tilsies å være en gauss-markov prosess. Stasjonærverdien til prosessen er beskrevet av autokorrelasjon funksjonen

$$\varphi_{XX}(\tau) = \sigma^2 e^{-T|\tau|} + m^2 \quad (3.5)$$

hvor korrelasjons tiden fortsatt er $\frac{1}{T}$, og spektraltetthet er gitt som

$$\phi_{XX}(\omega) = \frac{2\beta_1\sigma^2}{\omega^2 + \beta_1^2} \quad (3.6)$$

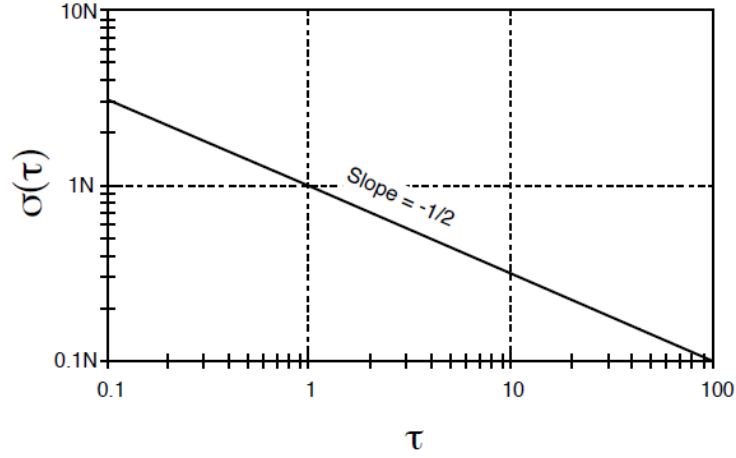
En 1.ordens gauss-markov prosess holder seg alltid innenfor et område hvor den "vandrør" fritt over tid. Et eksempel på en slik prosess vil bli illustrert senere.

3.2 Gyroparametre

I avsnittene under vil tre av gyroparametrene bli forklart.

3.2.1 Angular Random Walk (ARW)

ARW kommer fra hvit støy i målingene fra gyroskopet. I databladene til ulike gyroskop er ARW konstanten ofte oppgitt med benevnningen $^{\circ}/\sqrt{hr}$. STIM202 gir et ut-signal som er av typen $^{\circ}/s$, som er vinkelhastighet. Ved en integrasjon vil man følgelig få vinkelen. Under integrasjonen vil også hvitstøyen (ARW) bli integrert opp, slik at den blir forsterket. Etter integrasjonen kan en si at støyen er en random walk prosess [5]. En slik prosess, er i statistikken brukt for å beskrive en situasjon der resultatet av et system er drevet av tilfeldige, ukorrelererte "steg".



Figur 3.2: Log-log graf for allan variansen ved ARW [9].

ARW konstanten beskriver gjennomsnittlig avvik eller feil som vil oppstå når signalet integreres. Dette er feil som oppstår spesielt på grunn av støy i signalet, uavhengig av andre egenskaper som bidrar til feil i vinkelhastigheten, som for eksempel skalafaktorfeil, eller biasfeil. Denne feilen vil øke jo lengre man integrerer over tid, og det gir dermed en grunnleggende begrensning for alle vinkelmålinger som fremkommer ved integrasjon av signalet. Hvis en hvit støy feil blir integrert over tid, vil effekten den har på det integrerte resultatet være lik [mulig ref ?]

$$\sigma_{\theta}(t) = \sigma \sqrt{\delta t t} \quad (3.7)$$

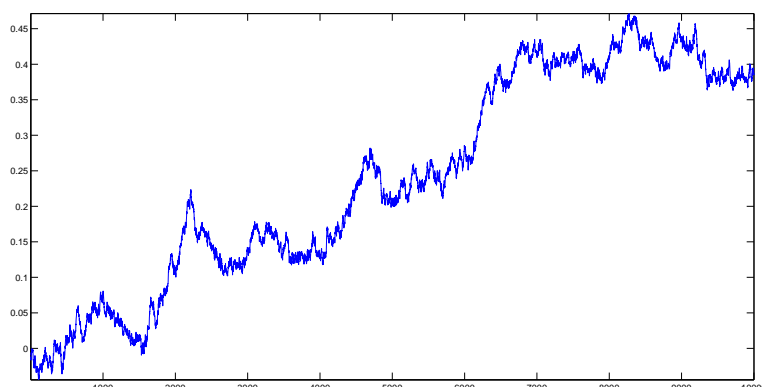
I et allan-varians plot kan denne konstanten leses av ved $\tau = 1$, som vist i figur 3.2 . En kan også finne bidraget til denne støykomponenten over en gitt tid (t). Hvis t for eksempel er 100 sekunder vil hvit støy bidraget fremkomme ved å multiplisere ARW-konstanten med $\sqrt{100}$ s. ARW konstanten er i et allan-varians plot definert som [9]

$$ARW = \sigma_{\theta}(1) \quad (3.8)$$

og feilen over tid er gitt av

$$Feil = ARW \sqrt{t} \quad (3.9)$$

hvor t er integrasjonstiden.



Figur 3.3: Eksempel på en RRW prosess.

3.2.2 Rate Random Walk (RRW)

En rate random walk prosess, som på godt norsk kan kalles en ”virrevandring-prosess”, sies å være av ukjent opphav [ref??]. En slik prosess kan beskrives som et begrenset tilfelle av eksponentielt korrelert støy, med svært lang korrelasjonstid. En slik prosess er også drevet av tilfeldige ukorrelerte steg, og den kan ”gå” forskjellige veier. Et eksempel på en RRW prosess kan sees i figur 3.3.

3.2.3 Bias instability

En annen koeffisient som er oppgitt i databladene er ”bias instability”. Denne parameteren representerer bunnpunktet i et Allan-varians plot, og sier noe om hvor mye man kan redusere feilen til gyroskopet, ved å ta gjennomsnitt av mange målinger. Hvis man deler opp måleserien i subgrupper med hensyn på tiden, og regner ut gjennomsnittet til disse, vil man nå et slikt bunnpunkt. Det er vanlig å nå dette punktet når måledataene er delt inn i 9 subgrupper, noe som krever en måleserie på ca 2.5 timer.

3.3 Simuleringsmodell som genererer målinger

Når en har opparbeidet kunnskap om støytypene og parametrene som inngår i gyrosignaler er det lettere å modellere disse matematisk. I det følgende vil det bli gjort modelleringer som på best mulig måte skal gjenspeile gyrosignalene fra gyroskopene STIM202 og SAR100.

For en tre-akset gyro er den generelle målelikningen definert som

$$\tilde{\underline{\omega}}_g^{ig} = \underline{\omega}_g^{ig} + \underline{\mu}^g + \underline{\beta}^g + \underline{v}^g \quad (3.10)$$

Komponentene som inngår i likningen ser slik ut på vektorform

$$\underline{\omega}_g^{ig} = [\omega_{g,x}^{ig}; \omega_{g,y}^{ig}; \omega_{g,z}^{ig}] \quad (3.11)$$

For en en-akset gyro, med for eksempel x-akse vil dette se slik ut

$$\omega = \omega_{g,x}^{ig} \quad (3.12)$$

$$\tilde{\omega} = \tilde{\omega}_{g,x}^{ig} \quad (3.13)$$

$$\mu = \mu_x^g \quad (3.14)$$

$$\beta = \beta_x^g \quad (3.15)$$

Gyromålingene for x-aksen modelleres deretter som

$$\tilde{\omega}_k = (1 + \tau)\omega_k + \mu_k + \beta_k + v_\omega \quad (3.16)$$

τ som beskriver skalafaktoren tas ikke med i dette tilfellet. μ og β beskrives av differensiallikningene

$$\dot{\mu} = -\frac{1}{T}\mu + v_\mu \quad (3.17)$$

Dette er differensiallikningen til en 1.ordens gauss-markov prosess som er definert tidligere i kapitlet i likning 3.4. Likningen modellerer farget støy i gyrosignalet. Den andre differensiallikningen er

$$\dot{\beta} = v_\beta \quad (3.18)$$

som beskriver støyprosessen rate random walk. De andre komponentene i 3.16 er ω_k som representerer sann vinkelfart og v_ω som representerer støyprosessen angular random walk.

Utifra beskrivelsen over kan en utlede en tilstandsrommodell for støyen, som har formen

$$\dot{\underline{x}} = [\dot{\beta} \quad \dot{\mu}]^T, \quad \underline{x} = [\beta \quad \mu]^T, \quad \underline{v}_\omega = \underline{v}_k \quad (3.19)$$

Denne tilstandsromformen kan igjen beskrives som et kontinuerlig system med en tilhørende diskret målelikning

$$\dot{\underline{x}} = F\underline{x} + G\underline{v} \quad (3.20)$$

$$\underline{z}_k = H\underline{x}_k + \underline{w}_k \quad (3.21)$$

som på matriseform kan skrives som

$$\dot{x} = \begin{bmatrix} 0 & 0 \\ 0 & -\frac{1}{T} \end{bmatrix} \begin{bmatrix} x_1 & x_2 \end{bmatrix}^T \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v_1 & v_2 \end{bmatrix}^T \quad (3.22)$$

$$\underline{z}_k = \begin{bmatrix} 1 & 1 \end{bmatrix} \underline{x}_k + \underline{w}_k \quad (3.23)$$

De andre systemmatrisene er :

$$\hat{P}_0 = \begin{bmatrix} (\frac{\pi}{180})^2 & 0 \\ 0 & (\frac{\pi}{180})^2 \end{bmatrix} \quad Q = \begin{bmatrix} (\frac{\pi}{180})^2 & 0 \\ 0 & \frac{2P_0(2,2)}{T} \end{bmatrix} \quad R_d = (\frac{\pi}{180})^2 \quad (3.24)$$

3.3.1 Diskretisering

$\dot{\underline{x}} = F\underline{x} + L\underline{u} + G\underline{v}$ diskretiseres på følgende måte:

$$x_{tk+1} = e^{F(tk+1-tk)} \underline{x}_{tk} + \int_{tk}^{tk+1} e^{F(tk+1-\tau)} L\underline{u}(\tau) d\tau + \int_{tk}^{tk+1} e^{F(tk+1-\tau)} G\underline{v} d\tau \quad (3.25)$$

$$\underline{x}_{k+1} = \Phi \underline{x}_k + \Lambda \underline{u}_k + \Gamma \underline{v}_k \quad (3.26)$$

hvor

$$\Phi = e^{F\Delta t} \quad (3.27)$$

og

$$\Lambda \underline{u}_k = \int_{tk}^{tk+1} e^{F(tk+1-\tau)} L\underline{u}(\tau) d\tau \quad (3.28)$$

siden det skal simuleres et gyroskop som ligger i ro vil pådragsmatrisa Λ være lik 0. Det er først når gyroskopet utsettes for en rotasjon eller dreining, at denne matrisen vil få verdier annet enn 0.

$$\Gamma = S = \Gamma Q \Gamma^T = \int_{t_k}^{t_{k+1}} \Phi G \tilde{Q} G^T \Phi^T d\tau \quad (3.29)$$

Hvis en antar likningen i 3.29, kan Γ og Q spaltes på følgende måte:

$$S = U D U^T \quad (3.30)$$

hvor U og D finnes ved hjelp av cholesky-faktorisering. Siden $S \geq 0$, vil $D \geq 0$ og diagonal på formen

$$D^{\frac{1}{2}} = \begin{bmatrix} \sqrt{(d_{11})} & & 0 \\ & \ddots & \\ 0 & & \sqrt{(d_{nm})} \end{bmatrix} \quad (3.31)$$

Videre lar man $Q=I$, og man kan da regne ut S og Γ på følgende måte:

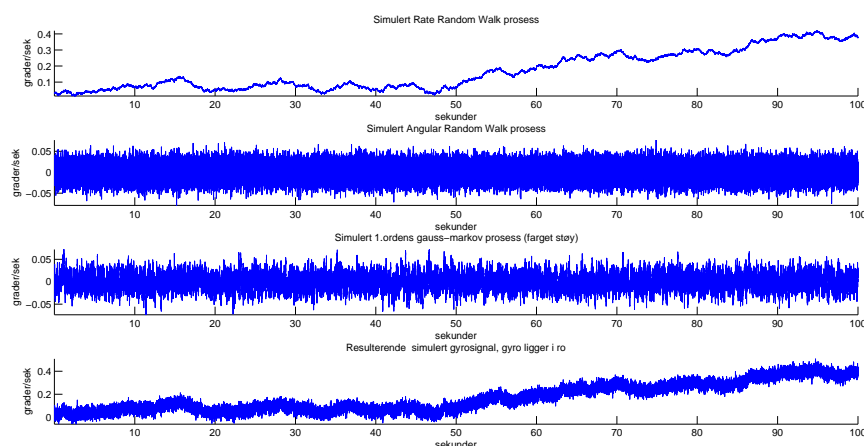
$$S = U D U^T = U D^{\frac{1}{2}} * I * D^{\frac{1}{2}} U^T \quad (3.32)$$

$$\Gamma = U D^{\frac{1}{2}} \quad (3.33)$$

I Matlab brukes følgende funksjoner i diskretiseringen

$$\Gamma = \text{kp2dpGa}(F, G, Q, T_s)$$

3.3.2 Simulering av gyroskopet STIM202

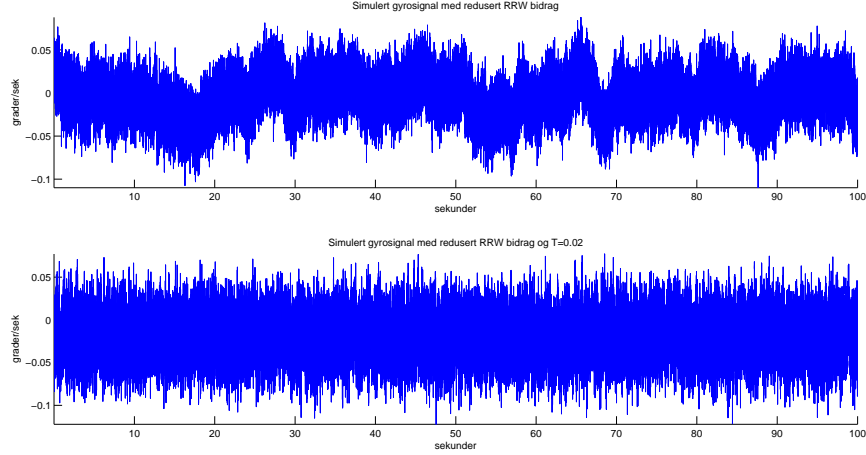


Figur 3.4: Simulert ARW prosess, RRW prosess, farget støy (1.ordens markov prosess) og resulterende simulert gyrosignal.

Figuren over viser en simulert 1. ordens gauss-markov prosess, en rate random walk prosess, en angle random walk prosess og det resulterende gyrosignalet fra simuleringsmodellen. Pseudokode for simuleringen kan sees i vedlegg A. Det kan sees at RRW prosessen har en tydelig drift i seg. Det er simulert i 100 sekunder med samplingstid $T_s = 0.001$, og en korrelasjonstid $T = 2$. Det kommer også tydelig frem at en random walk prosess kan drifte mer og mer ettersom tiden går. Det resulterende simulerte gyrosignalet fra de tre prosessene er vist nederst i figur 3.4 Det er tatt utgangspunkt i et en-akset gyroskop, som ligger i ro. Det vil si sann vinkelfart, $\omega_k = 0$. Det simulerte gyrosignalet har litt drift, noe som skyldes bidraget fra RRW prosessen og den fargede støyen. For at gyroparametrene skal kunne bli estimert på en optimal måte, er det viktig at det simulerte gyrosignalet speiler det faktiske gyrosignalet fra de nevnte gyroskopene. Den matematiske modellen vil da være tilpasset det reelle signalet på en best mulig måte.

Signalet fra gyroskopet STIM202 ser ut til å ha veldig liten drift. For at man skal kunne etterligne dette signalet i simuleringsmodellen, er det viktig at bidraget fra RRW prosessen blir redusert betydelig. Det kan også hende at korrelasjonstiden, T , må varieres. Man vil på denne måten forhåpentligvis bli kvitt uønskede svingninger og ujevnheter i signalet. Plotet i figur 3.5 viser to simulerte gyrosignal hvor noen parametre er ”skrudd på” i hvert tilfelle.

Det øverste signalet inneholder et redusert bidrag fra RRW prosessen og en korrelasjonstid, $T = 2$. Det er tydelig at det fortsatt er litt svingnin-



Figur 3.5: Simulert gyrosignal med redusert bidrag fra ARW prosess, samt mindre korrelasjonstid ,T.

ger i signalet, selv etter at dette er gjort. Det nederste signalet har samme reduksjon i RRW, men en korrelasjonstid $T = 0.02$. Dette resulterer i et stabilt signal uten store svingninger. For å få en optimal visuell likhet mellom simulert og reell gyrodata, må en gjøre det simulerte signalet litt bredere. Simuleringsmodellen for gyroskopet STIM202 inneholder følgende matriser med konstanter som er forandret fra den opprinnelige simuleringsmodellen

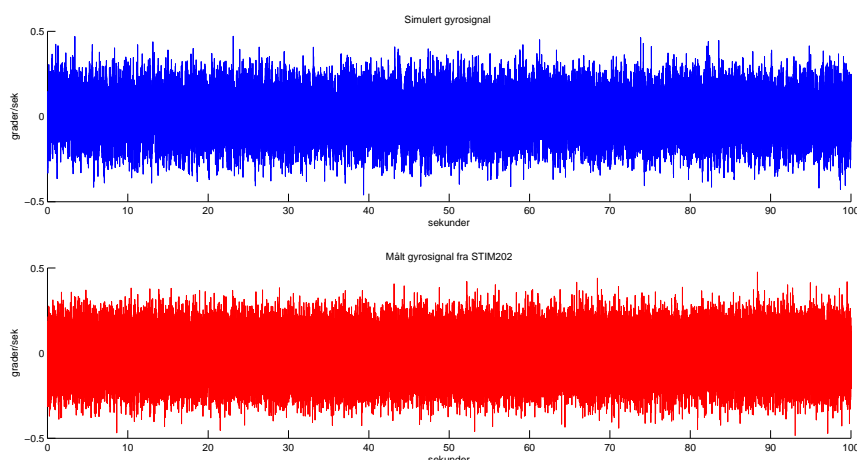
$$\hat{P}_0 = \begin{bmatrix} (\frac{5*\pi}{180})^2 & 0 \\ 0 & (\frac{5*\pi}{180})^2 \end{bmatrix} Q = \begin{bmatrix} 1e^{-10} & 0 \\ 0 & \frac{2P_0(2,2)}{T} \end{bmatrix} T = 0.02 \quad R_d = (\frac{5 * \pi}{180})^2 \quad (3.34)$$

Hvis en til slutt sammenligner målt gyrodata fra STIM202 gyroskopet, med signalet fremkommet av simuleringsmodellen ser man at de er veldig visuelt like. Dette er vist i figur 3.6

Det har nå blitt utviklet en tilfredsstillende matematisk modell for signalet fra gyroskopet STIM202. Denne modellen vil bli brukt i blant annet maximum likelihood estimeringen.

3.4 Simulering av gyroskopet SAR100

Plot av et målt signal fra SAR100 er vist i figur 3.7. I figuren er det plotet et vanlig målt signal, og et signal hvor middelverdien er trukket fra i ettertid. Dette har blitt gjort for å fjerne bias som er i signalet. Det kan se ut som



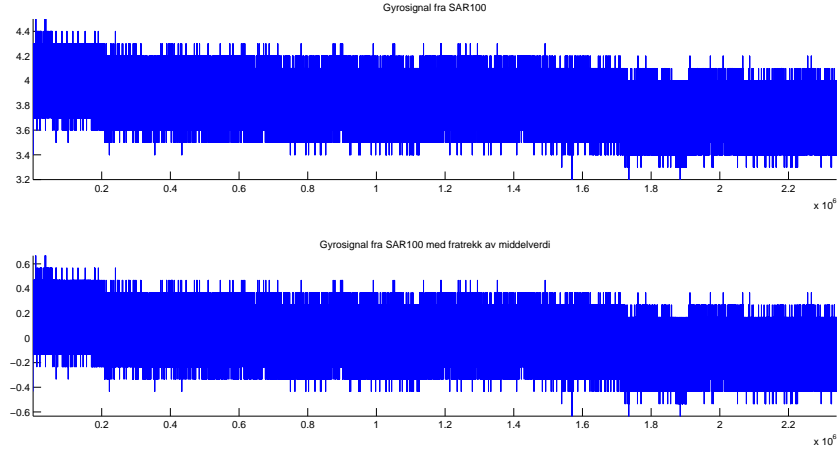
Figur 3.6: Simulert og målt signal.

om signalet er ”kuttet” på toppene. Det har også betydelig dårligere oppløsning enn STIM202. Signalet er gradvis synkende med tiden, og det kan tenkes at dette er vanskelig å etterligne visuelt. På bakgrunn av dette vil den vanlige simuleringsmodellen bli brukt når signalet skal estimeres i maximum likelihood metoden, men modellen vil inneholde et betydelig RRW bidrag.

3.5 Allan variansen

Allan variansen er navngitt etter David W. Allan, og ble først brukt til å si noe om stabiliteten til en oscillator. Allan variansen kan brukes til å karakterisere underliggende tilfeldige prosesser, som gir opphav til støy på målinger av et signal [1]. Metoden er derfor effektiv når det skal identifiseres kilder til forskjellige typer av støy i målinger. Likheten mellom en oscillator og et gyroskop, har ført til at denne metoden er mye brukt til å karakterisere støy og drift i en rekke forskjellige MEMS-sensorer. Det er også viktig å bemerke seg at allan variansen estimerer stabilitet på grunnlag av støyprosesser, uten å ta hensyn til systematiske feil og ujevnheter som for eksempel drift i frekvens, eller temperaturpåvirkninger.

Fordelen med denne typen varians framfor vanlig varians er at den konvergerer for de fleste forekommende typer av støy, hvor den vanlige variansen ikke alltid konvergerer til en endelig verdi. ”Flicker noise” og random walk støy er to eksempler på tilfeller der den klassiske variansen ikke konvergerer. Med klassisk varians menes [7]



Figur 3.7: Målt gyrosignal fra SAR100 med og uten bias.

$$VAR(X) = E[(X - \mu)^2] \quad (3.35)$$

Resultatet av allan variansen er av IEEE standarden [9] relatert til de fem forskjellige støykildene, ARW, RRW, bias instability, quantization noise og rate ramp. De to første støytypene er også oppgitt i databladene til gyroskop generelt. De tre første av disse støytypene har blitt forklart i detalj i avsnittene over.

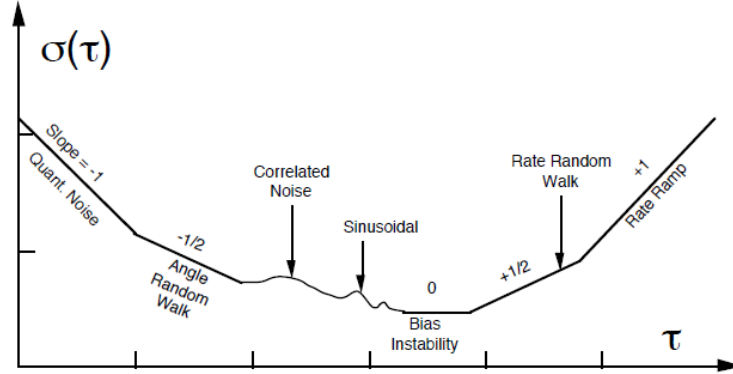
Allan variansen fremkommer av N sampler fra et gyroskop med en samplingstid τ_0 . Samplene deles deretter inn i grupper med størrelse $\tau_0, 2\tau_0, \dots, k\tau_0$ der $k < (\frac{N}{2})$. Hver gruppe har et tilhørende gjennomsnitt av antall sampler i gruppen dividert på lengden til gruppen. Allan variansen er derfor en funksjon av gjennomsnittstid og den er definert som [9]

$$\sigma^2(\tau) = \frac{1}{2} \langle (\bar{\Omega}_{k+m} - \bar{\Omega}_k)^2 \rangle = \frac{1}{2\tau^2} \langle (\theta_{k+2m} - 2\theta_{k+m} + \theta_k)^2 \rangle \quad (3.36)$$

hvor $\langle \rangle$ er ensemble gjennomsnittet og τ er lengden på gruppene. Allan variansen kan bli uttrykt ved vinkelfarten $\Omega(t)$ fra gyroen, eller ved vinkelen gitt ved

$$\theta(t) = \int^t \Omega(t') dt' \quad (3.37)$$

Estimatet av allan variansen fremkommer av følgende formel



Figur 3.8: Log-log graf for allanvariansen med alle feilkilder [9].

$$\sigma^2(\tau) = \frac{1}{2\tau^2(N-2m)} \sum_{k=1}^{N-2m} (\theta_{k+2m} - 2\theta_{k+m} + \theta_k)^2 \quad (3.38)$$

Allan variansen gir et godt bilde på hvor de forskjellige støytypene er mest fremtredende i et signal. Nøkkeltall for disse støytypene kan også leses av i et log log plot av $\sqrt{\sigma^2(\tau)}$ som vist i figuren over.

3.5.1 Oppsummering

Som sagt er det mange støytyper i et gyro signal. Disse er fremtredende ved ulike regioner av (τ) . Gyroparametrene som er beskrevet i avsnittene over har konstanter forbundet til seg. Disse konstantene er et mål på hvor nøyaktig en gyro, eller en annen MEMS-sensor vil være. Det er da snakk om nøyaktigheten i målingene til disse sensorene. Konstantene er også som regel å finne i sensorenes tilhørende datablader. I figur 3.8 er det vist et typisk Allan varians plot. Ved hjelp av slike plot er det enkelt å identifisere de forskjellige støybidragene i et signal. Hvis det kan antas at alle støybidragene er statistisk uavhengige, kan allan variansen uttrykkes ved enhver τ som summen av disse

$$\sigma_{tot}^2(\tau) = \sigma_{ARW}^2(\tau) + \sigma_{RRW}^2(\tau) + \sigma_{BiasInst}^2(\tau) + \dots \quad (3.39)$$

3.6 Maximum Likelihood estimering

Det er ønskelig å estimere gyroparametrene ved hjelp av maximum likelihood estimering. I statistikken er dette en metode for å estimere parametre i en statistisk modell. Ved å anvende denne estimeringsmetoden på en måleserie, med en tilhørende statistisk modell, vil maximum likelihood estimatet gi et anslag for modellens parametre. Metoden vil da velge parameter-verdier som produserer en distribusjon, som gir de observerte data størst sannsynlighet. Følgelig vil en da maksimalisere sannsynlighetsfunksjonen. Maximum likelihodd metoden er allsidig og den er gjeldende for de fleste modeller, og til ulike typer av data. I tillegg gir den effektive metoder for å kvantifisere usikkerheten til anvendte måleserier.

3.6.1 Prinsipp

Anta at et sample x_1, x_2, \dots, x_n kommer fra n uavhengige og identisk fordelte observasjoner, med en tilhørende fordeling med ukjent tetthet $f_0(\cdot)$. Det er også antatt at funksjonen f_0 tilhører en bestemt type av fordelinger $\{f(\cdot|\theta), \theta \in \Theta\}$, også kalt den parametriske modellen, slik at $f_0 = f(\cdot|\theta_0)$ [3]. Verdien θ_0 er ukjent, men blir omtalt som den ”sanne” verdien til parametervektoren. Det er ønskelig å finne en estimator $\hat{\theta}$ som vil være så nær den sanne verdien (θ_0) som mulig.

For å bruke maximum likelihood metoden må en først spesifisere en felles tetthetsfunksjon for alle observasjoner. For et sample som kommer fra en uavhengig, og identisk fordeling av observasjoner, vil denne tetthetsfunksjonen se slik ut

$$f(x_1, x_2, x_3, \dots, x_n|\theta) = f(x_1|\theta)f(x_2|\theta)f(x_3|\theta) \quad (3.40)$$

Deretter lar en de observerte verdiene $x_1, x_2, x_3, \dots, x_n$ være faste ”parametre” av funksjonen, mens θ vil være funksjonvariabelen og kan dermed variere [8]. En kan nå kalle fordelingsfunksjonen for sannsynligheten (likelihood)

$$\mathcal{L}(\theta|x_1, \dots, x_n) = f(x_1, x_2, \dots, x_n|\theta) = \prod_{k=1}^n f(x_k|\theta) \quad (3.41)$$

I praksis er det mer passende å arbeide med logaritmen til sannsynlighetsfunksjonen kalt log-likelihood, eller et gjennomsnitt av denne

$$\ln \mathcal{L}(\theta|x_1, \dots, x_n) = \sum_{k=1}^n \ln f(x_k|\theta) \quad (3.42)$$

$$\hat{E} = \frac{1}{n} \ln \mathcal{L} \quad (3.43)$$

Maksimum likelihood metoden estimerer θ_0 ved å finne en verdi av θ som maksimaliserer $\hat{E}(\theta|x)$. Denne estimeringsmetoden blir kalt maksimum likelihood estimator (MLE) av θ_0 , og kan uttrykkes på formen

$$\hat{\theta}_{mle} = \operatorname{argmax} \hat{E}(\theta|x_1, \dots, x_n) \quad (3.44)$$

Et MLE-estimat er det samme uansett om man maksimaliserer fordelingsfunksjonen (sannsynligheten), eller log-likelihood funksjonen, siden log er en monoton transformasjon. MLE estimatoren er for mange modeller en eksplisitt funksjon av de observerte dataene x_1, \dots, x_n . For noen modeller vil maksimaliseringen ikke ha noen løsning eller være ukjent. Løsninger av MLE-estimerer kan også finnes numerisk ved hjelp av optimaliseringsmetoder. Disse kan for eksempel løses i Matlab.

3.7 Kalmanfilter

Innen statistikken er kalmanfilteret en matematisk metode navngitt etter Rudolf E. Kalman. Filteret bruker støyfylte måleserier for å produsere verdier som er nærmere de ”sanne” verdiene av målingene. Dette gjøres ved at filteret predikterer en verdi og estimerer den tilhørende usikkerheten til denne verdien. Deretter regnes det ut et vektet gjennomsnitt av den predikterte verdien og den målte verdien, hvor den verdien med minst usikkerhet blir vektet mest. På denne måten har estimatene som blir produsert av filteret en tendens til å være nærmere de ”sanne” målingene enn de opprinnelige målingene. Dette er fordi det vektete gjennomsnittet har en bedre estimert usikkerhet enn de inngående verdiene til estimatet.

På bakgrunn av disse egenskapene er kalmanfilteret ofte brukt som en tilstandsestimator for estimering av tilstander i lineære dynamiske systemer. Filteret er rekursivt, og det trenger kun forrige tidsskritt-verdi av en tilstand, for å beregne en ny tilstand. På denne måten behøver man ikke kunnskap om alle tidsskrittene til hver av tilstandene for å beregne neste tidsskritt. Kalmanfilteret eksisterer både for kontinuerlige og diskrete systemer. Som sagt tidligere er det ønskelig å uttrykke den matematiske modellen på formen, Hallingstad 2005 [6]

$$\dot{\underline{x}}(t) = F(t)\underline{x}(t) + L(t)\underline{u}(t) + G(t)\underline{v}(t) \quad (3.45)$$

$$\underline{z} = H\underline{x} + \underline{\omega} \quad (3.46)$$

hvor

$$\begin{aligned} E\{\underline{x}(t_0)\} &= \bar{\underline{x}}_0 & E\{(\underline{x}(t_0) - \bar{\underline{x}}_0)(\underline{x}(t_0) - \bar{\underline{x}}_0)^T\} &= \bar{P}_0 & E\{\underline{x}(t_0)\underline{v}^T(t)\} &= \underline{0} \\ E\{\underline{v}(t)\} &= \underline{0} & E\{\underline{v}(t)\underline{v}^T(\tau)\} &= \delta(t - \tau)\tilde{Q}(t) & E\{\underline{x}(t_0)\underline{w}_k^T\} &= \underline{0} \\ E\{\underline{w}_k\} &= \underline{0} & E\{\underline{w}_k\underline{w}_l^T\} &= \delta_{kl}R_k & E\{\underline{v}(t)\underline{w}_k^T\} &= \underline{0} \end{aligned}$$

3.7.1 Kontinuerlig Kalmanfilter

Følgende likninger inngår for beregning av tilstander, kovarians og kalmanfilterforsterkning i det kontinuerlige tilfellet

$$\dot{\hat{\underline{x}}}(t) = F(t)\hat{\underline{x}}(t) + L\underline{u}(t) + K(t)(\underline{z}(t) - H(t)\hat{\underline{x}}(t)), \hat{\underline{x}}(t_0) \text{ gitt} \quad (3.47)$$

$$\dot{\hat{P}}(t) = F(t)P(t) + \hat{P}(t)F^T(t) + G(t)\tilde{Q}(t)G^T(t) - K(t)\tilde{R}(t)K^T(t), \hat{P}(t_0) \text{ gitt} \quad (3.48)$$

$$K(t) = \hat{P}(t)H^T(t)\tilde{R}^{-1}(t) \quad (3.49)$$

3.7.2 Diskret Kalmanfilter

Følgende likninger inngår i det diskrete tilfellet av kalmanfilteret:

Tidsoppdatering:

$$\bar{\underline{x}}_{k+1} = \Phi_k\hat{\underline{x}}_k + \Lambda_k\underline{u}_k, \bar{\underline{x}}_0 \text{ gitt} \quad (3.50)$$

$$\bar{P}_{k+1} = \Phi_k\hat{P}_k\Phi_k^T + \Gamma_kQ_k\Gamma_k^T \quad (3.51)$$

Måleoppdatering:

$$\hat{\underline{x}}_k = \bar{\underline{x}}_k + K_k(\underline{z}_k - H_k\bar{\underline{x}}_k) \quad (3.52)$$

$$K_k = P_k H_k^T (H_k \bar{P}_k H_k^T + R_k)^{-1} \quad (3.53)$$

$$\hat{P}_k = (I - K_k H_k) \bar{P}_k \quad (3.54)$$

der

\bar{x}_k : A priori estimert tilstandsvektor

\hat{x}_k : A posteriori estimert tilstandsvektor

\bar{P}_k : Kovariansen til a priori estimert tilstandsvektor

\hat{P}_k : Kovariansen til a posteriori estimert tilstandsvektor

K_k : Kalmanfilterforsterkning

Q_k : Prosesstøyens kovarians

R_k : Målestøyens kovarians

z_k : Måling

3.7.3 Kontinuerlig diskret Kalmanfilter

I det kontinuerlige diskrete kalmanfilteret er det en blanding av kontinuerlige og diskrete likninger:

Tidsoppdatering:

$$\dot{\hat{x}}(t) = F(t)\bar{x}(t) + Lu(t), := \hat{x}_k \quad (3.55)$$

$$\dot{\hat{P}}(t) = F(t)\bar{P}(t)F^T(t) + G(t)\tilde{Q}(t)G^T(t), \bar{P}(t_k) := \hat{P}_k \quad (3.56)$$

Måleoppdatering:

$$\hat{x}_k = \bar{x}_k + K_k(z_k - H_k \bar{x}_k), \bar{x}_k := \bar{x}(t_k) \quad (3.57)$$

$$K_k = \bar{P}_k H_k^T (H_k \bar{P}_k H_k^T + R_k)^{-1}, \bar{P}_k := \bar{P}(t_k) \quad (3.58)$$

$$\hat{P}_k = (I - K_k H_k) \bar{P}_k \quad (3.59)$$

3.8 Stokastiske variabler

En stokastisk variabel er en variabel som får sin numeriske verdi bestemt utifra et tilfeldig eksperiment. I motsetning til en deterministisk prosess hvor det bare er en mulighet for hvordan prosessen utvikler seg over tid, vil en

stokastisk prosess ha en viss ubestemthet i dens fremtidige utvikling. Denne beskrivelsen passer flere av støyprosessene i et gyrosignal, som for eksempel en RRW prosess.

Utviklingen av slike prosesser er beskrevet av sannsynlighetsfordelinger. Selv om startpunktet for prosessen er kjent er det fortsatt mange veier prosessen kan "gå". Det enkleste tilfellet av en stokastisk prosess er det diskrete tilfellet hvor en sekvens av tilfeldige variabler utgjør prosessen, som for eksempel en markov prosess.

Stokastiske variable kan enten være diskrete eller kontinuerlige. Mange av støyprosessene i gyrosignaler kan tilskrives å være slike prosesser.

3.9 Sannsynlighetsfordeling

Sannsynlighetsfordeling er mye brukt til å beskrive hvordan stokastiske variable og andre tilfeldige utvalg vil fordele seg. Utfallet av en stokastisk variabel kan ikke forutsies, men en tilhørende sannsynlighetsfordeling vil beskrive sannsynligheten for at de forskjellige utfallene skal inntre for variabelen. De underliggende egenskapene som bestemmer en sannsynlighetsfordeling er forskjellige. Man kan for eksempel tilskrive sannsynlighetsfordelinger til en serie av terningkast, eller høyden på mennesker i et samfunn.

Hensikten med en sannsynlighetfordeling er at den skal tildele en sannsynlighet til hvert intervall $[a,b]$ av mulige reelle tall \mathbb{R} . Alle stokastiske variable, X , har en tilhørende sannsynlighetsfordeling som tildeler en sannsynlighet $P[a \leq X \leq b]$ i intervallet $[a,b]$. Dette er sannsynligheten for at X har en verdi innenfor det gitte intervallet. Sannsynlighetsfordelingen $F(x)$ tilhørende X , kan beskrives som [5]

$$F(x) = P[X \leq x] \quad (3.60)$$

for alle x inneholdt i \mathbb{R} . Verdiene i funksjonen vil variere fra 0 til 1. Sannsynlighetsfordelinger har også en tilhørende sannsynlighetstetthetsfunksjon $f(x)$. Denne er definert som [5]

$$f(x) = \frac{dF(x)}{dx} \quad (3.61)$$

Dette er en statistisk funksjon som sier hvordan tettheten til observasjonene i en populasjon er fordelt. Sannsynlighetstettheten $f(x)$ til en variabel x tilfredsstiller at $f(x) \geq 0$ for alle x , hvor det totale arealet under kurven er lik 1. Kontinuerlige og diskrete sannsynlighetsfordelinger beskrives på forskjelli-

ge måter. Sannsynligheten for at x , i det kontinuerlige tilfellet, er i intervallet $[a, b]$ er [5]

$$P(a \leq x \leq b) = \int_a^b f(x) dx \quad (3.62)$$

Siden fordelingen over er beskrevet i det kontinuerlige tilfellet, er den gitt som et integral istedenfor en sum. I det diskrete tilfellet blir sannsynligheten summert på en annen måte. Her er funksjonen gitt som summen av en rekke endelige sprang, noe som også karakteriserer X som en diskret stokastisk variabel gitt av

$$F(x) = P(X \leq x) = \sum_{x_i \leq x} p(x_i) \quad (3.63)$$

Likning 3.63 tydeliggjør at diskrete sannsynlighetsfordelinger er summen av sannsynlighetene for en rekke enkeltutfall.

Den desidert viktigste fordelingen blant sannsynlighetsfordelinger er normalfordelingen. Dette er mye på grunn av sentralgrenseteoremet. Teoremet sier at en sum av uavhengige, og identisk fordelte tilfeldige variabler, vil gå mot en normalfordeling når antallet går mot ∞ (ref, wiki-11/05). Denne egenskapen fører til at mange hendelser i naturen og samfunnet kan beskrives av normalfordelingen, på en god måte. Normalfordelingen beskrives av sannsynlighetstetthetsfunksjonen

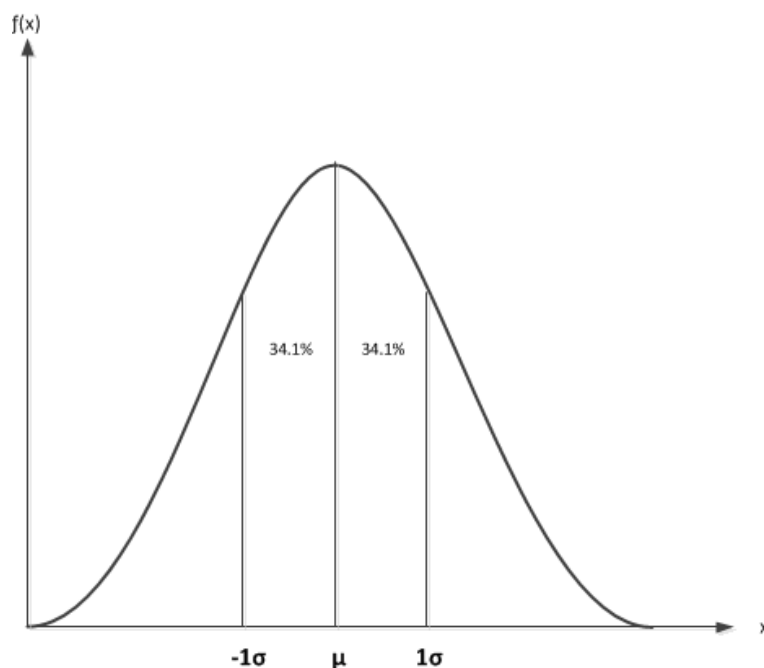
$$f(x) = \frac{1}{\sigma^2 \sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \quad (3.64)$$

hvor $\mu = E\{x\}$ = forventningsverdi og $\sqrt{\sigma^2} = \sqrt{E\{(x-\mu)^2\}}$ = standardavvik til fordelingen [7]. Dette er de to parametrene som bestemmer spredningen. Formen på grafen til fordelingen er "bjelleaktig". Dette er fordi tetthetsfunksjonen inneholder en eksponentialfunksjon, noe som også resulterer i en symmetrisk kurve. En illustrasjon av dette er vist i figur 3.9.

3.10 MLE anvendt på gyromålinger

Som sagt tidligere benytter MLE metoden seg av sannsynlighetstetthetsfunksjoner for å utarbeide estimer.

En kan først anta måledata fra gyroen, $Z_N = \{z_1, z_2, \dots, z_k, \dots, z_N\}$. Den tilhørende sannsynlighetstetthetsfunksjonen er da $p(\underline{z} : \underline{p})$, hvor \underline{p} er fullstendig ukjent [4]. Det er ønskelig å identifisere nøkkelparametre som ligger til



Figur 3.9: Normalfordelingen.

grunn for produksjon av en spesifikk måleserie, Z_N , fra gyroskopet. Videre er gyromodellen definert etter likningene 3.20 og 3.21, og diskretiseringen av de inngående matrisene. Av dette kan en derfor anta at matrisene F, Q, R_d og \hat{P}_0 inneholder ukjente elementer av en parametervektor \underline{p} .

For å beregne et estimat av de underliggende parametrene til gyromålingene, må en maksimisere funksjonen $p(Z_N:\underline{p})$ eller beregne logaritmen til denne, $\log p(Z_N:\underline{p}) = J(N:\underline{p})$. MLE estimatet av $\underline{\hat{p}}$ vil da endre verdien av \underline{p} som igjen maksimaliserer loglikelihood funksjonen $J(N:\underline{p})$:

$$J(k+1:\underline{p}) = J(k:\underline{p}) + \log \mathcal{N}(\underline{\bar{z}}(k+1:\underline{p}), P_z(k+1:\underline{p})), \quad J(0:\underline{p}) = 0 \quad (3.65)$$

Både $\underline{\bar{z}}(k+1:\underline{p})$ og $P_z(k+1:\underline{p})$ kan beregnes ved hjelp av målinger, og et kalmanfilter. Initialbetingelsen $\underline{p}=\underline{p}_0$ gjør at en kan tolke parametervektoren som kjent og variablene kan dermed uttrykkes ved [4]

$$\underline{\bar{z}}(k+1:\underline{p}) = H\bar{x}_{k+1} \quad (3.66)$$

og

$$\bar{P}_z(k+1 : \underline{p}) = H\bar{P}_{k+1}H^T + R \quad (3.67)$$

Matrisen \bar{P} og vektoren \bar{x} , fremkommer av kalmanfilter likningene beskrevet tidligere. Det er ønskelig å tilegne måledataene til en kjent statistisk sannsynlighetsfordeling. På denne måten kan man anta en kjent form på funksjonen $p(\underline{z}:\underline{p})$. Ut-signalet fra gyroskopene kan bli beskrevet av normalfordelingen. Siden parametervektoren \underline{p} inneholder flere parametre p_i , vil det passe å bruke den multivariate formen av normalfordelingen. Det generelle uttrykket for denne fordelingen er:

$$p(\bar{x}) = \frac{1}{(2\pi)^{\frac{d}{2}}|\Sigma|^{\frac{1}{2}}} e^{-\frac{1}{2}(\bar{x}-\bar{\mu})^T\Sigma^{-1}(\bar{x}-\bar{\mu})}, \mathcal{N}(\bar{\mu}, \Sigma) \quad (3.68)$$

hvor $\bar{\mu} = E\{\bar{x}\} = [\mu_1, \mu_2, \dots, \mu_d]^T$ og $\Sigma = E\{(\bar{x} - \bar{\mu})(\bar{x} - \bar{\mu})^T\}$, som har matriseformen

$$\Sigma = \begin{bmatrix} \sigma_{11} & \sigma_{12} & \dots & \sigma_d \\ \vdots & & & \\ \sigma_{d1} & \dots & \dots & \sigma_{dd} \end{bmatrix}$$

I matrisen beskriver diagonalelementene varians, og ikke-diagonalelementene beskriver kovarians. Denne fordelingen er lik normalfordelingen i likning 3.64, men den gjelder for n dimensjoner av Σ og μ . Tar man utgangspunkt i likning 3.21 med antagelsen om at \underline{u} er normalfordelt med middelerdi lik 0, og en kovariansmatrise \bar{P}_z , har man

$$\mathcal{N}(\bar{z}(k+1 : \underline{p}), \bar{P}_z(k+1)) = \frac{e^{-\frac{1}{2}(\underline{z}_{k+1} - \bar{z}_{k+1})^T \bar{P}_z(k+1:\underline{p})^{-1}(\underline{z}_{k+1} - \bar{z}_{k+1})}}{(2\pi)^{\frac{n_z}{2}} |\bar{P}_z(k+1 : \underline{p})|^{\frac{1}{2}}} \quad (3.69)$$

For å bestemme maximum likelihood estimatet må man beregne $\ln p(Z_N:\underline{p}) = \ln J(N:\underline{p})$ som er gitt av likning 3.65. Tar man logaritmen av likning 3.69 og setter dette inn i 3.65 får man

$$\begin{aligned} J(k+1 : \underline{p}) &= J(k : \underline{p}) - \frac{n_z}{2} \ln(2\pi) - \frac{1}{2} \ln |\bar{P}_z(k+1 : \underline{p})| \\ &\quad - \frac{1}{2} (\underline{z}_{k+1} - \bar{z}_{k+1})^T \bar{P}_z(k+1 : \underline{p})^{-1} (\underline{z}_{k+1} - \bar{z}_{k+1}) \end{aligned} \quad (3.70)$$

I likningen over inngår det kun skalar verdier. Det er tatt utgangspunkt i log likelihoodfunksjonen da denne er mer behagelig å arbeide med. Videre vil en søkealgoritme i Matlab, iterere seg mot et minimumspunkt i funksjonen $J(k+1 : \underline{p})$, slik at det fremkommer et estimat av parametervektoren \underline{p} .

3.10.1 Implementering av parametervektoren

Størrelsen på parametervektoren \underline{p} blir bestemt av antall parametre man ønsker å estimere. I et gyrosignal er det særlig nøkkelparametre som sier noe om ytelsen og nøyaktigheten til gyroskopet som er av interesse. I tabell 2.1 er det gjort en oversikt over verdier for slike parametre, hentet fra databladene til gyroskopene. Målelikningen i 3.16 inneholder en slik parameter med notasjonen, v^g , som er en ren hvitstøyprosess.

En kan estimere slike parametre ved å karakterisere elementer som er forbundet med disse konstantene i den matematiske modellen. Denne er blant annet gitt av matrisene i 3.34. Andre parametre som vil bli tatt med i estimeringen er diagonal elementene i kovariansmatrisen \hat{P}_0 og korrelasjonstiden, T , i likning 3.17. Elementene i parametervektoren plasseres deretter på sin tilhørende plass i den matematiske modellen

$$F = F(p) = \begin{bmatrix} 0 & 0 \\ 0 & -\frac{1}{p(1)} \end{bmatrix} \hat{P} = \begin{bmatrix} p(2)^2 & 0 \\ 0 & p(3)^2 \end{bmatrix} Q = Q(p) = \begin{bmatrix} 1e-10 & 0 \\ 0 & \frac{2*p(3)^2}{p(1)} \end{bmatrix} R_d = p(4) \quad (3.71)$$

I den matematiske modellen til STIM202 er bidraget fra RRW prosessen ≈ 0 , og den vil derfor ikke inngå i parametervektoren. Ved å anvende en optimaliseringsrutine på et funksjonsscript, hvor \underline{p} er input i funksjonen, vil rutinen iterere seg til et minimum ved å starte i punktet p_0 . I denne optimaliseringsrutinen har man også mulighet til å avgrense søkeområdet til funksjonen. Kalmanfilteret med de implementerte estimeringsalgoritmene, er integrert i funksjonsscriptet. Pseudokode for dette kan sees i vedlegg A.

3.10.2 Test av MLE på simulert gyrodata

For å få kunnskap om identifikasjonsprosessen ble maximum likelihood metoden først testet på simulert gyrodata. Dette ble gjort slik at en kan sammenligne estimeringsresultatene (\underline{p}), med de faktiske verdiene i simuleringsmodellen. På denne måten får man verifisert hvilke av elementene i \underline{p} som oppnår konvergens.

I tabell 3.1 er det gjort en oversikt over første kjøring av maximum likelihood metoden anvendt på generert gyrodata fra simuleringsmodellen. I tabellen er det oppgitt sanne verdier av parametrene (\underline{p}_{sann}), startverdiene (\underline{p}_0), og de estimerte resultatene (\underline{p}). Elementene i \underline{p} er gitt av matrisene i 3.71. For at funksjonen skal kunne klare å konvergere seg mot de sanne verdiene (\underline{p}_{sann}) kan en avgrense søkeområdet til funksjonen. Dette søkeområdet er gitt av en øvre (ub) og en nedre (lb) grense som fremkommer av

$$lb = (0.1 * \underline{p}_{sann}) \quad (3.72)$$

$$ub = (10 * \underline{p}_{sann}) \quad (3.73)$$

Ved å bruke 0.1 og 10 som faktorer har funksjonen fortsatt et stort område å bevege seg på. Det er viktig at parametrene konvergerer i et slikt område, da dette betyr at funksjonen klarer seg ”på egenhånd”. På denne måten får man verifisert estimerbarheten til de forskjellige parametrene.

I det følgende er det presentert resultater fra estimeringen av det simulerte signalet til STIM202.

Forsøk 1

I forsøk 1 ble parametrene T , $\hat{P}(1, 1)$, $\hat{P}(2, 2)$ og R_d estimert. I tabell 3.1 kan en se at det bare er parameteren $p(4) = R_d$ som konvergerer mot sin sanne verdi, $p_{sann}(4)$. Det kan tenkes at $p(2) = \hat{P}(1, 1)$ ikke er observerbar. Det er også en mulighet for at denne parameteren påvirker observerbarheten til de andre parametrene i parametervektoren.

Tabell 3.1: Resultater av forsøk 1.

	Korrelasjonskoeffisient (T)	$\hat{P}(1, 1)$	$\hat{P}(2, 2)$	R_d
Initiell verdi (\underline{p}_0)	2.0	1.6	6.0	1.5
Estimert verdi (\underline{p})	10.0	20.0	8.9707	1.0049
Sann verdi (\underline{p}_{sann})	1.0	2.0	3.0	1.0

Forsøk 2

I forsøk 2 ble parametrene T , $\hat{P}(2, 2)$ og R_d estimert. For å påvise at $p(2)$ påvirker observerbarheten til andre parametre ble denne satt til verdien, $\hat{P}(1, 1) = 1$, i den matematiske modellen. Siden denne parameteren ble satt til en fast verdi, inneholder parametervektoren \underline{p} kun 3 parametre. Resultatet av dette kan sees i tabell 3.2. Av tabellen fremgår det at konvergensen nå er mye bedre for parametrene, T og $\hat{P}(2, 2)$, enn i forsøk 1. Fortsatt er estimatet av T et lite stykke fra sin opprinnelige verdi.

Tabell 3.2: Resultater av forsøk 2.

	Korrelasjonskoeffisient (T)	$\hat{P}(2, 2)$	R_d
Initiell verdi (p_0)	4	3	6
Estimert verdi (p)	2.828	2.210	2.990
Sann verdi (p_{sann})	2	2	3

Forsøk 3

I forsøk 3 ble også parametrene T , $\hat{P}(2, 2)$ og R_d estimert. For å bygge kunnskap om parameteren T , ble den tilhørende verdi økt til 5. Som følge av dette konvergerer den ikke like mye som i forsøk 2. Det ser ut til at korrelasjonskoeffisienten påvirker hvor langsomt eller raskt $\hat{P}(2, 2)$ konvergerer mot sin sanne verdi. Parameteren R_d ser derimot ut til å være upåvirket av dette, og det kan dermed konkluderes med at denne parameteren er godt observerbar. De estimerte verdiene kan sees under.

Tabell 3.3: Resultater av forsøk 3.

	Korrelasjonskoeffisient (T)	$\hat{P}(2, 2)$	R_d
Initiell verdi (p_0)	10.0	3.0	6.0
Estimert verdi (p)	8.04	2.40	2.991
Sann verdi (p_{sann})	5	2	3

Forsøk 4

I forsøk 4 ble de samme parametrene estimert. For å verifisere bedre konvergens av $\hat{P}(2, 2)$, ved lave verdier av T , ble denne satt til $T = 1$. Som antatt i forsøk 3, medfører dette at parameteren konvergerer mot sin sanne verdi. R_d er fortsatt upåvirket av endringene i de andre parametrene. Resultatene kan sees i tabell 3.4

Tabell 3.4: Resultater av forsøk 4.

	Korrelasjonskoeffisient (T)	$\hat{P}(2, 2)$	R_d
Initiell verdi (p_0)	2	3	9
Estimert verdi (p)	1.275	2.085	2.991
Sann verdi (p_{sann})	1	2	3

Forsøk 5

I forsøk 5 ble alle de sanne verdiene av parametrene endret. For å etterligne det reelle gyrosignalet som skal estimeres senere, ble det tatt utgangspunkt i modell 3.34 for valg av verdier. Tabell 3.5 viser at alle parametrene i parametervektoren konvergerer mot sine sanne verdier.

Tabell 3.5: Resultater av forsøk 5.

	Korrelasjonskoeffisient (T)	$\hat{P}(2, 2)$	R_d
Initiell verdi (p_0)	0.04	0.118	0.157
Estimert verdi (p)	0.0187	0.0773	0.0780
Sann verdi (p_{sann})	0.02	0.0785	0.0785

Forsøk 6

I siste forsøk var det bare R_d som ble estimert. De to andre parametrene ble fjernet fra parametervektoren og satt til faste verdier der, $T = 0.02$ og $\hat{P}(2, 2) = 1$. Ved å gjøre dette ble bidraget fra farget støy i det simulerte signalet satt ≈ 0 . Dette ble gjort for å se om de foregående parametrene, samt bidraget

fra farget støy i den matematiske modellen, hadde en eventuell påvirkning på estimatet av R_d . I tabell 3.6 kan det sees at R_d fortsatt konvergerer mot den samme verdien som tidligere.

Tabell 3.6: Resultater av forsøk 6.

	R_d
Initiell verdi (p_0)	6
Estimert verdi (p)	2.991
Sann verdi (p_{sann})	3

3.11 Autokorrelasjon

Autokorrelasjon er beskrevet som kryss-korrelasjonen av et signal med seg selv. På en enkel måte kan det sies at det er likheten mellom observasjoner, som en funksjon av tidsforskjellen mellom dem. Ved å anvende autokorrelasjon på et signal kan en finne repeterende mønstre, som for eksempel tilstedeværelsen av et periodisk signal, som har vært skjult under støy.

I statistikken beskriver autokorrelasjonen til en tilfeldig prosess korrelasjonen mellom verdier i prosessen ved forskjellige tidspunkt. Dette kan både uttrykkes som en funksjon av to tidspunkt og som en funksjon av tidsforskjellen. Korrelasjonskoeffisienten er beskrevet av [5]

$$\rho = \frac{E[XY] - E[X]E[Y]}{\sigma_X \sigma_Y} \quad (3.74)$$

Denne koeffisienten er et mål på hvilken grad av lineær avhengighet det er mellom X og Y. Det er da snakk om krysskorrelasjonen mellom disse. Hvis X og Y er uavhengige er ρ lik 0. Hvis Y er en lineær funksjon av X vil ρ være i området $[-1,1]$ hvor da 1 indikerer perfekt korrelasjon og -1 indikerer perfekt anti-korrelasjon. Denne lineære avhengigheten vil også gjelde hvis en korrelerer et signal med seg selv. Autokorrelasjonen til et signal x er gitt som [5]

$$\varphi_{xx}(t_1, t_2) = E[x(t_1)x(t_2)] = \int_{-\inf}^{\inf} dx_1 \int_{-\inf}^{\inf} dx_2 x_1 x_2 f_2(x_1, t_1; x_2, t_2) \quad (3.75)$$

Hvis det er en forskjell, τ , mellom t_1 og t_2 kan for eksempel t_2 skrives lik

$$t_2 = t_1 + \tau \quad (3.76)$$

Da vil $f_2(x_1, t_1; x_2, t_2)$ bli lik $f_2(x_1, t_1; x_2, t_1 + \tau)$ som er uavhengig av t_1 , men fortsatt en funksjon av τ . En kan da tilsi at korrelasjonsfunksjonen også er en funksjon av τ slik at

$$\varphi_{xx}(\tau) = E[x(t_1)x(t_1 + \tau)] \quad (3.77)$$

som har egenskapene [5]

$$\varphi_{xx}(0) = E[x^2] \quad (3.78)$$

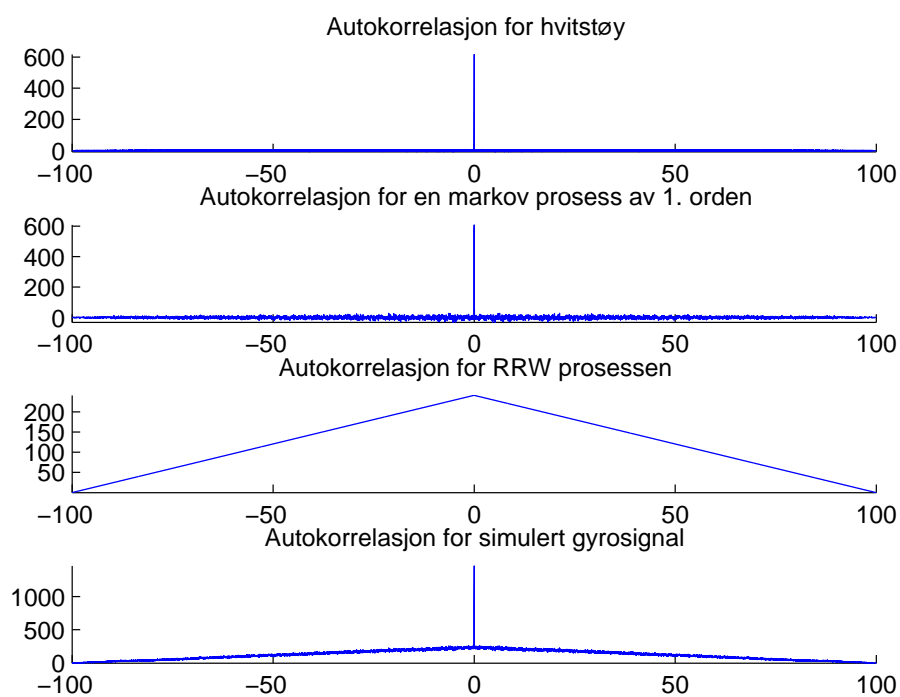
$$\varphi_{xx}(-\tau) = \varphi_{xx}(\tau) \quad (3.79)$$

$$\varphi_{xx}(0) \geq |\varphi_{xx}(\tau)| \quad (3.80)$$

Plot av autokorrelasjonen til hvitstøy, en 1.ordens markov prosess gitt av likning 3.17(farget støy), og en RRW prosess er vist i figur 3.10. At korrelasjon sier noe om avhengigheten mellom signaler kommer veldig tydelig frem her. For en hvitstøy prosess er det ingen avhengighet i autokorrelasjonen, og den er dermed ”spikerformet”. For en 1. ordens markov prosess er signalet mer avhengig av seg selv. Dette kommer tydelig frem i autokorrelasjonen til det simulerte signalet. I figur 3.11 kan det sees at kurven beveger seg langsomt mot 0. Det er den fargede støyen i signalet som bidrar til denne karakteristiske kurven. Avhengigheten kan også påvises i likningen for en 1. ordens markov prosess gitt av 3.17.

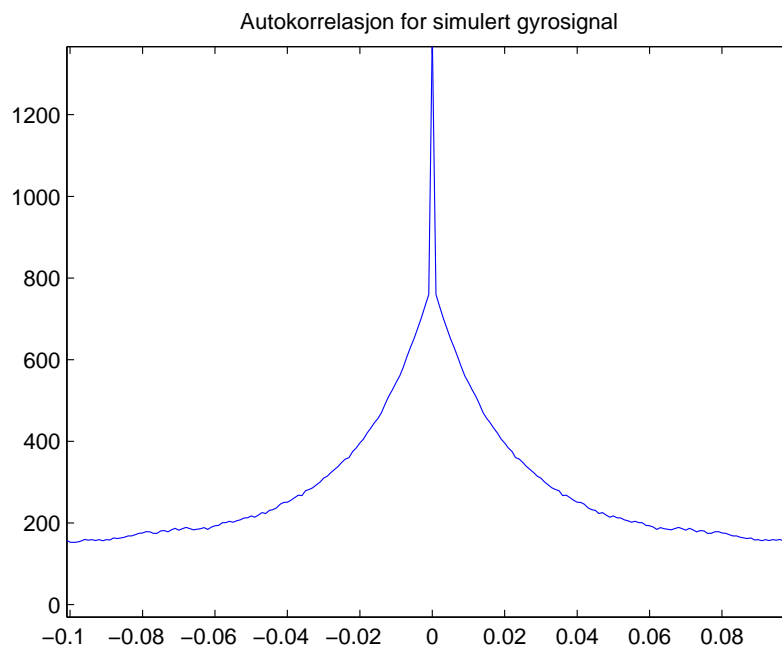
I figur 3.10 er det gjort simuleringer på 100 sekunder. Hver av prosessene kommer fra en og samme simulering. Den fargede støyen er simulert med en korrelasjonstid, $T = 0.02$. Siden autokorrelasjon bare er definert for stasjonære prosesser [finne ref], er ikke autokorrelasjonen til RRW prosessen gjeldende. Plotet av denne er heller ment som en illustrasjon i forhold til det simulerte signalet da det kan tenkes at den totale autokorrelasjonen er en sum av de andre autokorrelasjonene.

Figur 3.11 viser et nærmere bilde av autokorrelasjonen. Når man zoomer inn i denne figuren, blir detaljene i plotet mer tydelig. Denne kurven har også form som en eksponentialfunksjon. For sammenlikning er det vist et tilsvarende plot for autokorrelasjonen til hvitstøy i 3.12. Her er det tydelig at kurven beveger seg mot 0 i en mer eller mindre loddrett linje. Dermed er det tydelig ingen korrelasjon i dette tilfellet, da hvitstøy er ukorrelet. Følgelig er det heller ingen korrelasjonstid i denne prosessen. I figur 3.11 kan man gjennomføre en kurvetilpasning til den fallende kurven. Man kan da

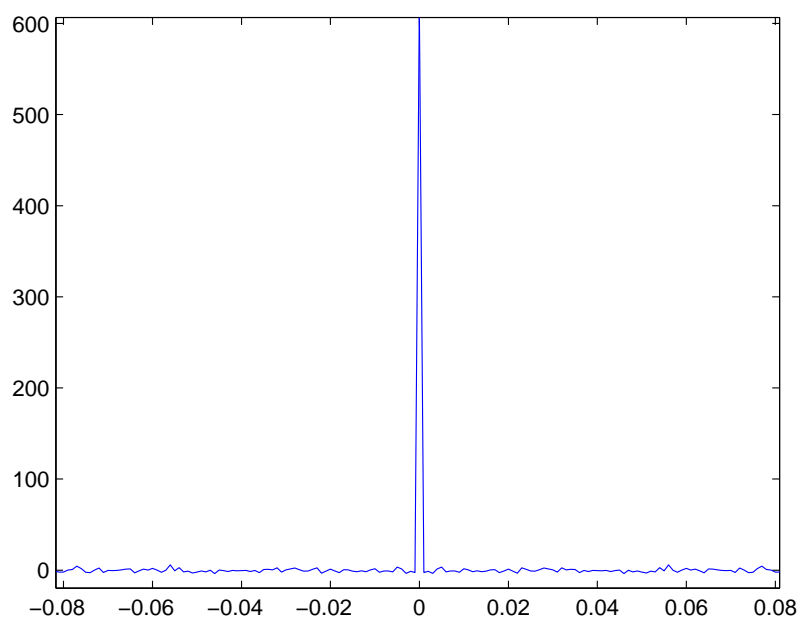


Figur 3.10: Autokorrelasjon simulert gyrosignal.

få tilpasset denne etter en kjent eksponentialfunksjon slik at tangenten til funksjonen kan illustreres. Denne tangenten vil skjære x-aksen i verdien av korrelasjonstiden, T , til prosessen.



Figur 3.11: Et nærmere bilde av autokorrelasjonen for simulert gyrosignal.



Figur 3.12: Et nærmere bilde av autokorrelasjonen for hvitstøy.

Kapittel 4

Resultater

I dette kapitlet vil resultatene for oppgaven bli presentert. Allanvarians resultatene vil bli presentert først. Deretter vil Maximum likelihood estimeringene, og til slutt autokorrelasjons resultatene.

4.1 Allanvarians

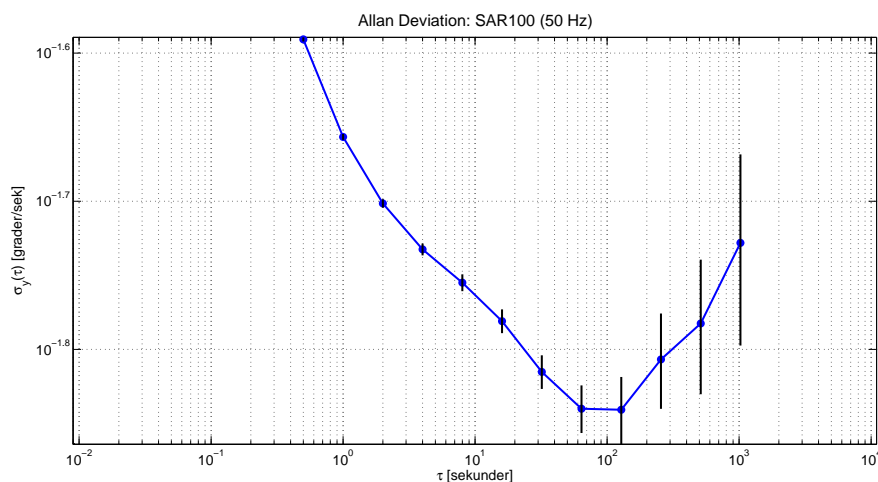
I det følgende vil allanvariansen til reell gyrodata bli presentert. Måling på gyroskopene er gjort etter beskrivelsen i underkapittel 2.3. Det har blitt samlet i forskjellige tidsforløp på de to gyroskopene. Temperaturen i omgivelsene har ligget på rundt 25°C.

4.1.1 SAR100

Allanvarians plot for aksen i SAR100 gyroskopet er vist i figur 4.1. Det ble gjort målinger på gyroskopet mens det lå i ro på et bord. Det antas at dette bordet er parallelt med horisontalplanet. Måleserien som ble brukt for å regne ut allanvariansen var på 13 timer/46800 sekunder.

Tabell 4.1: Resultater fra allan varians for SAR100.

	Angular Random Walk	Bias Instability
Datablad	$0.65 \text{ } ^\circ/\sqrt{h} = 0.010833 \text{ } ^\circ/\sqrt{s}$	$50 \text{ } ^\circ/h = 0.013889 \text{ } ^\circ/s$
Gyro-aksen	$0.0187 \text{ } ^\circ/\sqrt{s}$	$0.01443 \text{ } ^\circ/s$



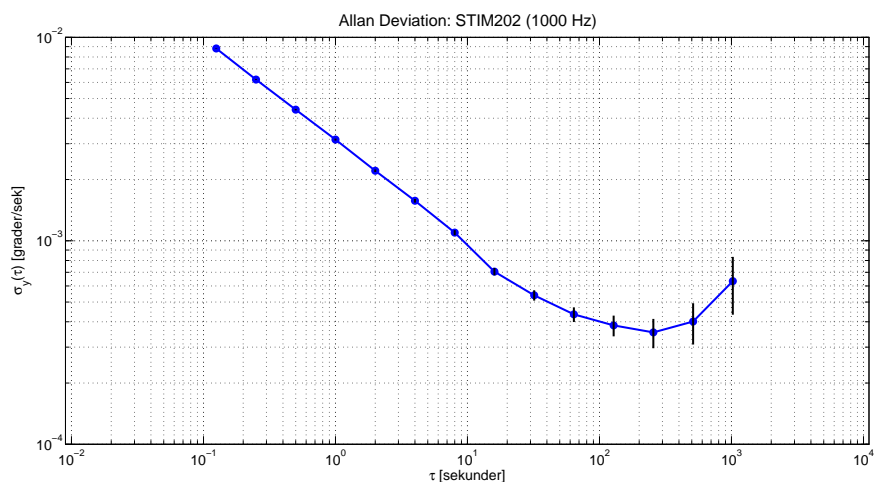
Figur 4.1: Allanvarians plot SAR100.

4.1.2 STIM202

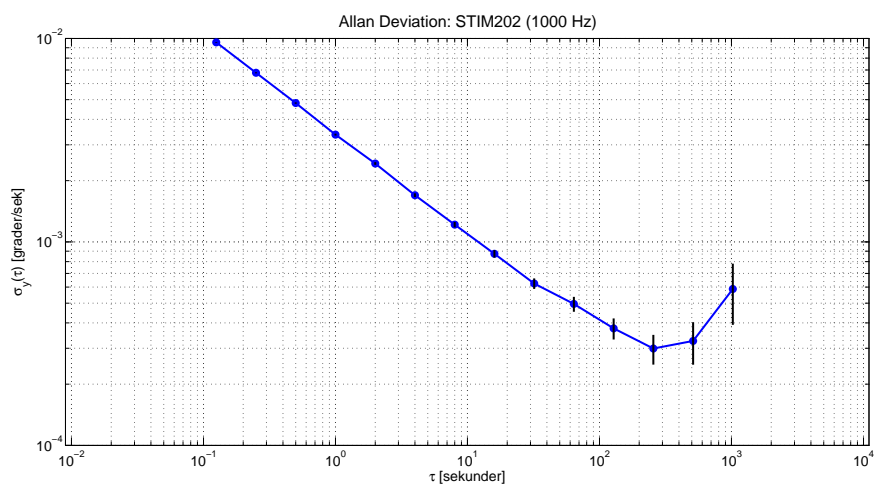
Allanvarians plot for z-aksen er vist i figur 4.2, for x-aksen i figur 4.3, og til slutt for y-aksen i figur 4.4. Gyrodata for de forskjellige aksene har fremkommet på lik måte. Måleseriene som ble brukt var 9000 sekunder lange. Som sagt tidligere leses ARW konstanten av i et allanvarians plot ved $\tau=0$. Verdien i punktet $\tau(0)$ har benevnningen $^{\circ}/\sqrt{s}$. En annen parameter som kan leses av i plotene er "bias instability". En oversikt av parametrene er gitt i tabell 4.2.

Tabell 4.2: Parametre for hver av aksene.

	Angular Random Walk	Bias Instability
Datablad	0.00333 $^{\circ}/\sqrt{s}$	0.0001388 $^{\circ}/s$
x-aksen	0.00333 $^{\circ}/\sqrt{s}$	0.0003 $^{\circ}/s$
y-aksen	0.0035 $^{\circ}/\sqrt{s}$	0.0005 $^{\circ}/s$
z-aksen	0.0032 $^{\circ}/\sqrt{s}$	0.00035 $^{\circ}/s$



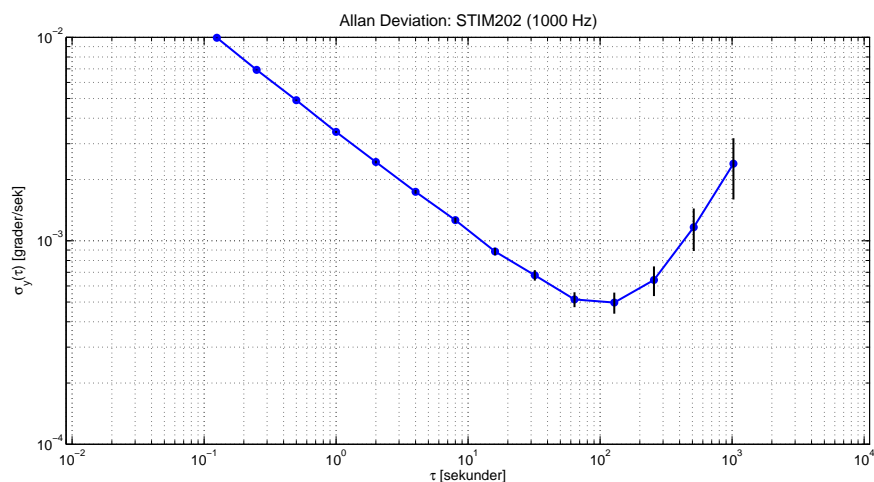
Figur 4.2: Allanvarians plot for z-aksen.



Figur 4.3: Allanvarians plot for x-aksen.

4.2 Maximum likelihood

I maximum likelihood metoden står en ganske fritt til å velge hvilke parametre som skal estimeres. Det er gjort estimering på begge gyroskopene med forskjellige parametre. For hver akse er det gjennomført to estimeringer. Det første estimatet inneholder en ren gjetting av verdiene på \underline{p}_0 , basert på verdier fra simuleringen. I det andre estimatet er p_0 gjettest utifra resultatene til den første estimeringen. Dette er gjort for at parametrene skal kunne oppnå bedre konvergens. I titlene til plotene og tabellene er det gjort rede for

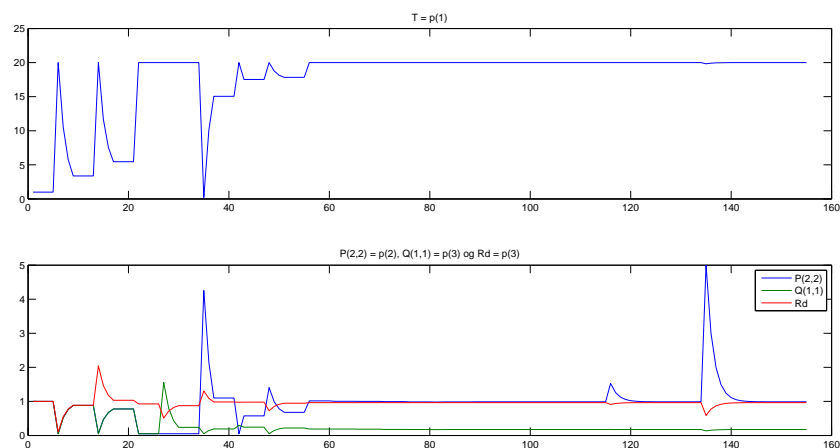


Figur 4.4: Allanvarians plot for y-aksen.

hvilken av de to gjetningene av p_0 som er brukt.

4.2.1 SAR100

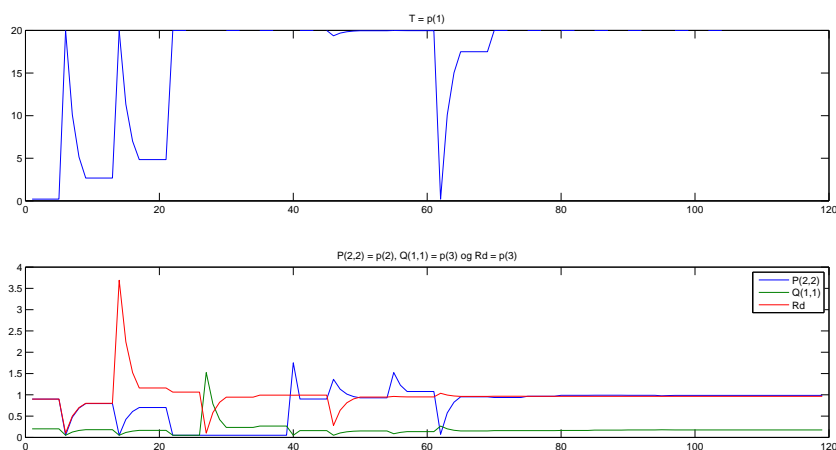
Estimerte parametre for SAR100 er T , $\hat{P}(2,2)$, $Q(1,1)$ og R_d . I hvert av resultatene er det gjort 100 000 gjennomganger av søkealgoritmen. Grensene for søkeområdet til algoritmen er gitt av lb som setter nedre grense, og ub som setter øvre grense. Resultatene kan sees i figur 4.5 og tabell 4.3.



Figur 4.5: Plot av konvergens (funksjonsvurdering) for parametrene til SAR100.

Tabell 4.3: Estimerte verdier for SAR100

	Korrelasjonskoeffisient (T)	$\hat{P}(2, 2)$	Q(1,1)	R_d
Initiell verdi (p_0)	1.0	1.0	1.0	1.0
Estimert verdi (p)	20.00	0.987653	0.174894	0.964548
lb	0.2	0.05	0.05	0.1
ub	20	5	5	10

Figur 4.6: Plot av konvergens (funksjonsvurdering) for parametrene til SAR100, med nye verdier av p_0 .

4.2.2 STIM202

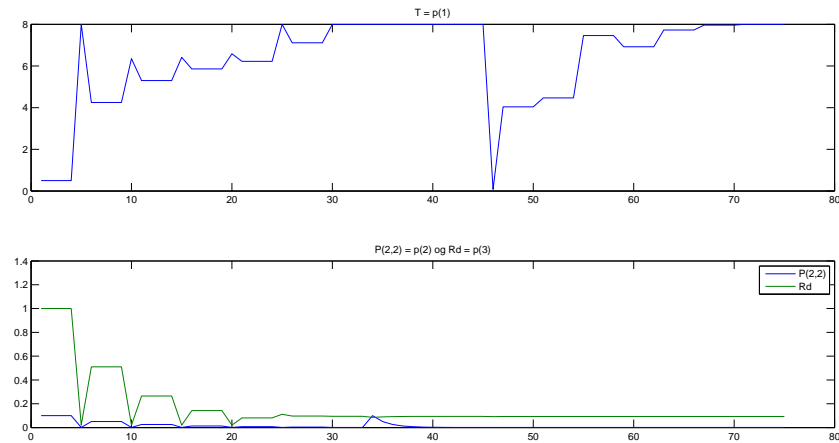
På bakgrunn av undersøkelsene i underkapittel 3.9.3, vil parametervektoren inneholde parametrene T , $\hat{P}(2, 2)$, og R_d . Det er gjort estimater for x , y og z -aksen. I hvert av resultatene er det gjort 100 000 gjennomganger av søkealgoritmen. Grensene for søkeområdet til algoritmen er gitt av lb som setter nedre grense, og ub som setter øvre grense. De estimerte resultatene er presentert under.

Tabell 4.4: Estimerte verdier til SAR100, med nye verdier av p_0 .

	Korrelasjonskoeffisient (T)	$\hat{P}(2,2)$	$Q(1,1)$	R_d
Initiell verdi (p_0)	0.02	0.9	0.2	0.9
Estimert verdi (p)	20.00	0.987825	0.1749594	0.9645881
lb	0.2	0.05	0.05	0.1
ub	20	5	5	10

Tabell 4.5: Estimerte verdier til z-aksen

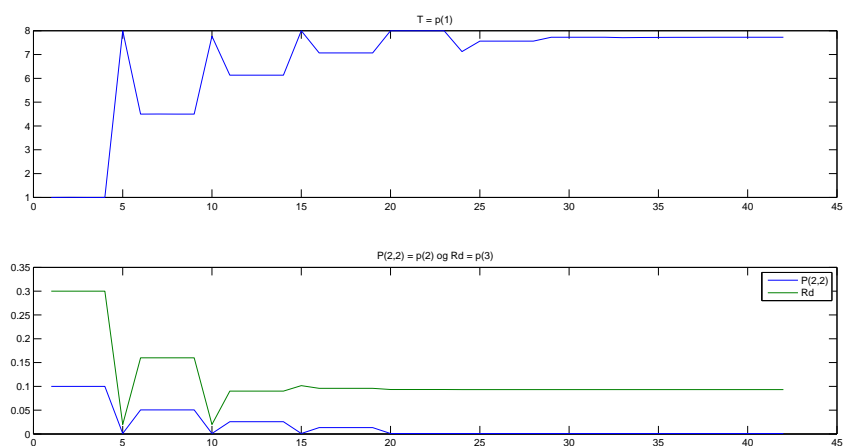
	Korrelasjonskoeffisient (T)	$\hat{P}(2,2)$	R_d
Initiell verdi (p_0)	0.5	0.5	1
Estimert verdi (p)	8.00	0.001	0.093262
lb	0.08	0.001	0.02
ub	8.0	0.1	2.0



Figur 4.7: Plot av konvergens (funksjonsvurdering) for parametrene for z-aksen

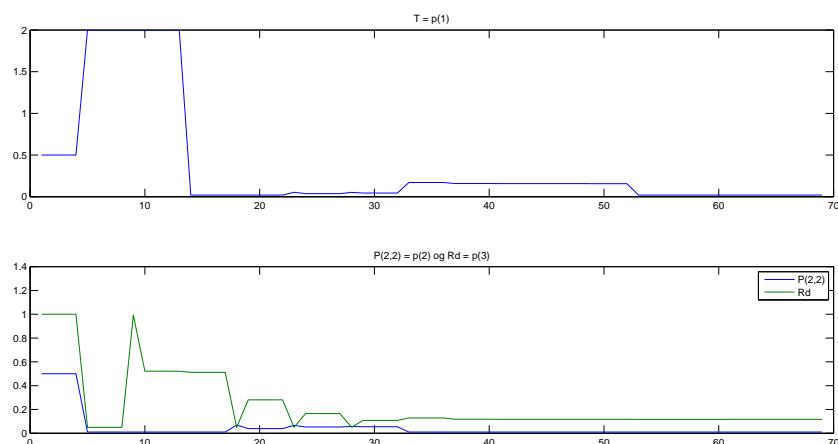
Tabell 4.6: Estimerte verdier til z-aksen, med nye verdier av (p_0)

	Korrelasjonskoeffisient (T)	$\hat{P}(2, 2)$	R_d
Initiell verdi (p_0)	1	0.1	0.3
Estimert verdi (p)	7.72481	0.001	0.093267
lb	0.08	0.001	0.02
ub	8.0	0.1	2.0

Figur 4.8: Plot av konvergens (funksjonsvurdering) for parametrene for z-aksen, med nye verdier av p_0

Tabell 4.7: Estimerte verdier til x-aksen

	Korrelasjonskoeffisient (T)	$\hat{P}(2, 2)$	R_d
Initiell verdi (p_0)	0.5	0.5	1
Estimert verdi (p)	0.02	0.01	0.117085
lb	0.02	0.01	0.05
ub	2.0	1.0	5.0



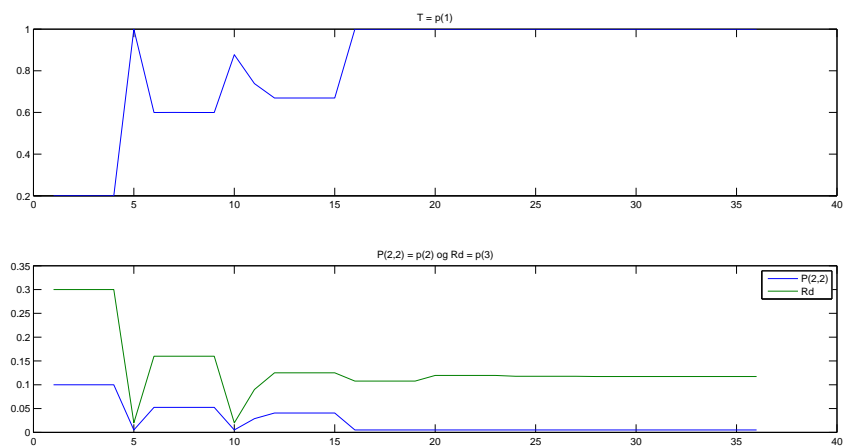
Figur 4.9: Plot av konvergens (funksjonsvurdering) for parametrene for x-aksen

Tabell 4.8: Estimerte verdier til x-aksen, med nye verdier av (p_0) .

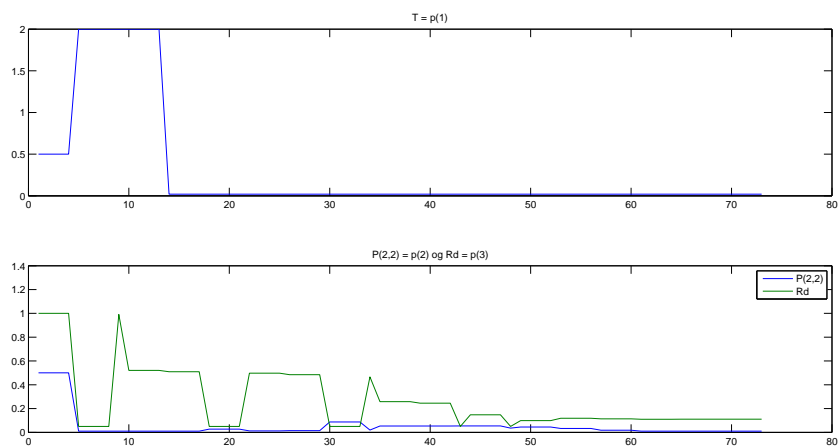
	Korrelasjonskoeffisient (T)	$\hat{P}(2, 2)$	R_d
Initiell verdi (p_0)	0.2	0.1	0.3
Estimert verdi (p)	1.00	0.005	0.1173080
lb	0.01	0.005	0.02
ub	1.0	0.5	2.0

Tabell 4.9: Estimerte verdier til y-aksen.

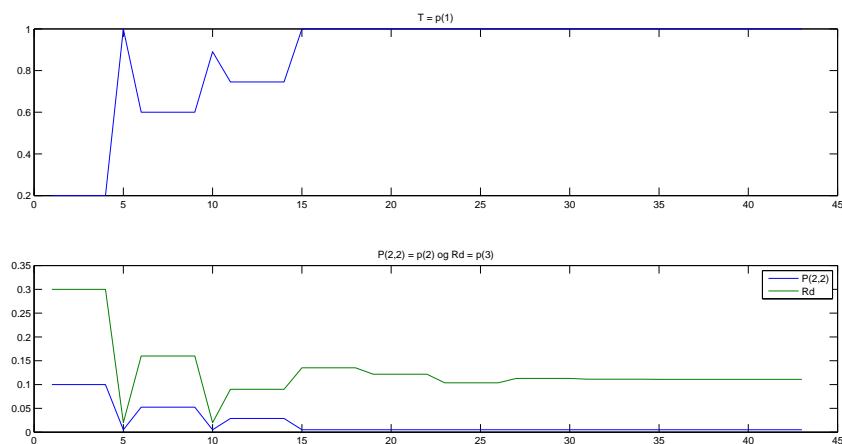
	Korrelasjonskoeffisient (T)	$\hat{P}(2, 2)$	R_d
Initiell verdi (p_0)	0.5	0.5	1.0
Estimert verdi (p)	0.02	0.01	0.11067971
lb	0.02	0.01	0.05
ub	2.0	1.0	5.0



Figur 4.10: Plot av konvergens (funksjonsvurdering) for parametrene for x-aksen, med nye verdier av (p_0) .



Figur 4.11: Plot av konvergens (funksjonsvurdering) for parametrene for y-aksen.

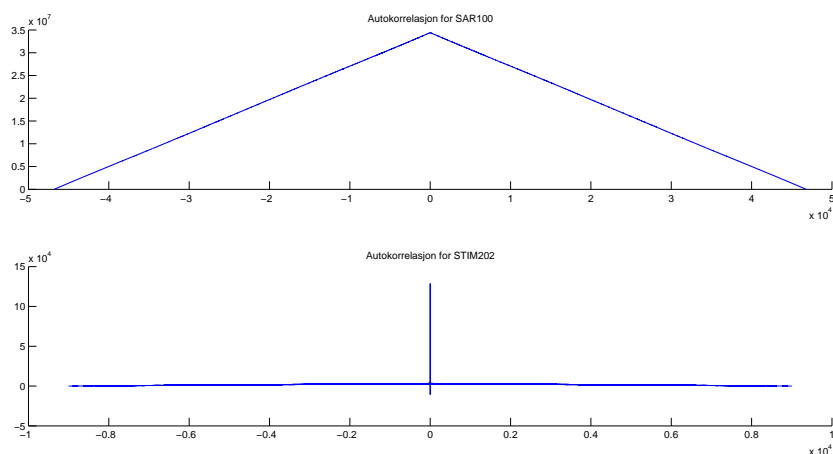


Figur 4.12: Plot av konvergens (funksjonsvurdering) for parametrene for y-aksen, med nye verdier av (p_0) .

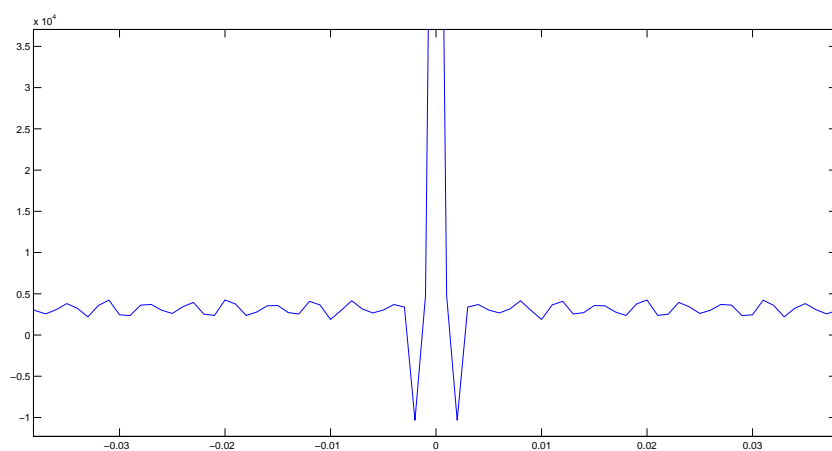
Tabell 4.10: Estimerte verdier til y-aksen, med nye verdier av (p_0) .

	Korrelasjonskoeffisient (T)	$\hat{P}(2, 2)$	R_d
Initiell verdi (p_0)	0.2	0.1	0.3
Estimert verdi (p)	1.00	0.005	0.1109981
lb	0.01	0.005	0.02
ub	1.0	0.5	2.0

4.3 Autokorrelasjon



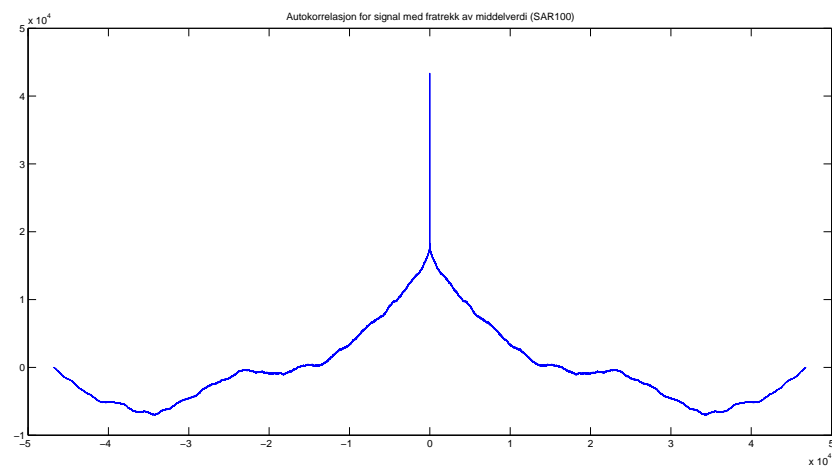
Figur 4.13: Autokorrelasjon for SAR100 og STIM202.



Figur 4.14: Et nærmere bilde av autokorrelasjonen til STIM202.

Ved å regne ut autokorrelasjonen til signalene kan en finne eventuelle korrelasjonstider T . Denne parameteren konvergente ikke i maximum likelihood metoden, og det er derfor nyttig å se hva som kan tolkes utifra autokorrelasjonen. I underkapittel 3.10 ble det gjort rede for en metode for å finne denne parameteren. Plot av autokorrelasjonen til SAR100 og STIM202 er presentert i figur 4.13.

Hvis man trekker fra middelveiden til SAR100 signalet, vil bias bli fjernet. Autokorrelasjon for dette signalet er vist i figur 4.15.



Figur 4.15: Autokorrelasjon av signal fra SAR100 uten bias.

4.4 Diskusjon

Utifra figur 4.13 kan autokorrelasjonen til SAR100 minne mye om autokorrelasjonen til RRW prosessen, som er fremstilt i figur 3.10. Grunnen til dette kan være den tydelige tilstedeværelsen av en RRW prosess i signalet til SAR100. Hvis en derimot trekker fra middelveiden til signalet slik at bias blir utlignet, ser autokorrelasjonen anderledes ut. Dette er vist i figur 4.15 og det kan sees en tydelig tilstedeværelse av farget støy.

Autokorrelasjonen for STIM202 er ganske lik autokorrelasjonen for hvitstøy, og man kan derfor anta at det er veldig liten eller ingen forekomst av farget støy i dette gyrosignalet.

I resultatene fra maximum likelihood metoden konvergente flere av estimatene for de forskjellige parametrene. R_d viste seg å være den eneste estimerbare verdien for STIM202. Etter å ha plotet autokorrelasjonen for signalet til STIM202, var det tydelig at det var mangel på farget støy i signalet. Dette forklarer også hvorfor parametrene, T og P(2,2), ikke lot seg estimere ved hjelp av denne metoden. Maximum likelihood metoden estimerte noen flere parametre fra signalet til SAR100. I tillegg til R_d , ble også parametrene $\hat{P}(2,2)$ og Q(1,1) estimert. Q(1,1) er et mål på RRW prosessen i signalet og det er derfor ikke unaturlig at estimatet for denne parameteren konvergerer mot en fast verdi. $\hat{P}(2,2)$ sier noe om nivået på den fargede støyen i signalet. Dette estimatet kan være nyttig hvis en skal modellere gyroskopet matematisk.

Allan variansen ga gode resultater for ARW estimatet til STIM202 og bias instability estimatet til SAR100. ARW estimatet til SAR100 og bias instability estimatet til STIM202 var derimot litt dårligere enn forventet.

Aksene i STIM202 ser ut til å ha ganske lik nøyaktighet, da hver av estimeringsmetodene ga relativt like resultater for hver akse. I tabell 4.11 er det vist en oversikt over de forskjellige estimeringsresultatene. Resultatene for hver av aksene til STIM202 er midlet, slik at hvert estimeringsresultat presenteres med kun en verdi.

Tabell 4.11: Oppsummering av estimerte verdier.

Parameter	Maximum likelihood	Allan variansen	Datablad
R_d	0.107191	-	-
ARW	-	$0.003344 \text{ } ^\circ/\sqrt{s}$	$0.00333 \text{ } ^\circ/\sqrt{s}$
Bias Instability	-	$0.00038333 \text{ } ^\circ/s$	$0.0001388 \text{ } ^\circ/s$

Kapittel 5

Konklusjon

I denne oppgaven har det blitt vurdert forskjellige estimeringsmetoder. Hver av metodene har gitt tilfredsstillende resultater av estimatene. Ved å kombinere kunnskapen om disse, kan man identifisere mange parametre som ligger til grunn for et spesifikt gyrosignal.

Estimering av gyromparametre ved bruk av allanvarians ga både gode, og mindre gode resultater, for hver av gyroskopene. For STIM202 var det stor likhet mellom estimerte ARW verdier og verdien i databladet. De estimerte verdiene av bias instability var derimot litt høyere enn verdiene i databladet. For SAR100 var det også likhet mellom den estimerte verdien av bias instability og verdien i databladet. ARW estimatet ble derimot litt avvikende.

En av grunnene til forskjellen mellom estimatene og databladet kan tenkes å være miljøforskjeller ved måling på gyroskopene. En annen grunn kan være at måleseriene til STIM202 er for korte til å nå det "sanne" bunnpunktet i allan plotet. SAR100 har stor likhet mellom estimert verdi og datablad. Den tilhørende måleserien som ble anvendt var 13 timer lang, mens måleseriene som ble brukt fra STIM202 var på ca 2,5-3 timer. En ulempe ved bruk av allan variansen er det faktum at det trengs lange måleserier for å oppnå gode resultater.

Maximum likelihood metoden resulterer i gode estimater av de observerbare verdiene i signalene. En fordel med denne metoden er at man kan velge hvilke parametre som skal estimeres. På denne måten får en også verifisert observerbarheten av parametre. Estimatene av parametrene til SAR100 og STIM202 konvergente mot sine "sanne verdier". Disse estimatene er ikke direkte sammenlignbare med databladet eller estimatene fra allan variansen. Hvis man bruker verktøyet som er utviklet i denne oppgaven på forskjellige gyroskoper, kan man gjøre seg en mening om hva som er "gode" verdier tilknyttet til gyroparametre.

Kapittel 6

Videre arbeid

Denne oppgaven kan fungere som et godt grunnlag for en eventuell implementering av de nevnte gyroskopene i et TNS. Utifra den matematiske modellen og Kalamanfilteret, kan en fremstille feilutviklingen til de forskjellige bidragene i signalet over tid. Da dette er gjort kan det også lages tilsvarende feilutviklinger for TNS systemet. På denne måten kan en opparbeide stor kunnskap om systemer, noe som kan føre til forbedret ytelse og nøyaktighet. En fortsettelse av oppgaven kan derfor gå ut på å implementere et slikt system.

Ettersom nøyaktigheten i MEMS-gyroskopene øker, kan det tenkes at retningen på ”sann nord” kan realiseres ved hjelp av et slikt gyroskop. Tidligere har dette ikke latt seg gjennomføre på grunn av for dårlig oppløsning og nøyaktighet i gyrosignalene til slike enheter. Det er usikkert om STIM202 kan oppfatte noe av jordrotasjonen når det ligger i ro.

En annen fortsettelse på oppgaven kan være å realisere ”sann nord” ved hjelp av STIM202 eller andre fremtidige nyinnkjøpte gyroskop. For å verifisere hvilke gyroskop som egner seg til en slik oppgave, kan man bruke arbeidet som er gjort i denne oppgaven som en eventuell dokumentasjon på dette. Man vil da få flere mål på rådata fra gyroskopet, i tillegg til et datablad.

Bibliografi

- [1] http://en.wikipedia.org/wiki/allan_variance (20/05-2011). Kilder kan sees med linken.
- [2] http://en.wikipedia.org/wiki/coriolis_effect (20/05-2011). Kilder kan sees med linken.
- [3] http://en.wikipedia.org/wiki/maximum_likelihood#cite_note-0 (20/05-2011). Kilder kan sees med linken.
- [4] System identification: 14.3 maximum likelihood : Gaussian model, general case (s.434-442). Kopi fra Oddvar Hallingstad.
- [5] Arthur Gelb. *Applied Optimal Estimation*. Second edition, 1974.
- [6] Oddvar Hallingstad. Standardmodeller og kalmanfilterlikninger. 2005. Notat til faget UNIK4500 Stokastiske systemer.
- [7] John Haugan. *Formler og tabeller*, volume 9. NKI Forlaget, 1 edition, 1974.
- [8] Simon Haykin. *Adaptive Filter Theory*. Fourth edition, 2002.
- [9] Haiying Hou Naser El-Sheimy and Xiaoji Niu. Analysis and modeling of inertial sensors using allan variance. *IEEE transactions on instrumentation and measurement*, 57, 2008.
- [10] Martin Lim Steven Nasiri and Mike Housholder. A critical review of the market status and industry challenges of producing consumer grade mems gyroscopes. *InSense*, 2011.
- [11] David H. Titterton and John L. Weston. *Strapdown Inertial Navigation Technology, volume 207*. Second edition, 2005.

- [12] N. Jeremy Usdin. Discrete simulation of colored noise and stochastic processes and l/f^α power law noise generation. *Proceedings of the IEEE*, vol 83, NO 5. The Institute of Electrical and Electronics Engineers, Inc., 1995.

Tillegg A

Pseudokode

A.1 Simulering

global N Z p_p k_p

Initialisering :

$\omega_k = v_k = \text{randn}('seed', n);$

$Z = \text{zeros}(N);$

$P_t = \text{zeros}(N); P_{x1} = \text{zeros}(N); P_{x2} = \text{zeros}(N); P_z = \text{zeros}(N);$

$x_0 \sim \mathcal{N}(0, \hat{P}_0);$

$T_s = 0.001$

$T = 2;$

Diskretisering :

$\Phi = e^{FT_s};$

$\Gamma = \text{ohkp2dpGa}(F, G, Q, T_s);$

$R_d = (\frac{\pi}{180})^2;$

$Q_d = I;$

Simulering :

$\hat{P}_0 = \bar{P}$

for $k = 0$ to N

$x = \Phi x + \Gamma v;$

$\underline{z}_k = Hx + \sqrt{R_d}w_k$

$\bar{P} = \Phi \bar{P} \Phi^T + \Gamma \Gamma^T$

$S = H \bar{P} H^T + R_d$

$P_t(k) = kT_s$

$P_{x1}(k) = [x_1; \sqrt{\bar{P}(1,1)}; -\sqrt{\bar{P}(1,1)}]$

$P_{x2}(k) = [x_2; \sqrt{\bar{P}(2,2)}; -\sqrt{\bar{P}(2,2)}]$

$P_z(k) = [Z_k; \sqrt{S}; -\sqrt{S}]$

end

IMatlab :
 $\Phi = Fi;$
 $\Gamma = Ga;$
 $w_k = v_k = randn();$
 $\underline{z}_k = Z(1, k);$
 $\hat{P}_0 = Pe0;$
 $\bar{P} = P_p;$
 $P_t = PA_t;$
 $P_{x1} = PA_x1;$
 $P_{x2} = PA_x2;$
 $P_z = PA_z;$

A.1.1 Kall av funksjon

$p_{true} = [T \sqrt{\hat{P}_0(1, 1)} \sqrt{\hat{P}_0(2, 2)} \sqrt{R_d}]$
 $p_0(1) = p_{true}(1) * 1.5$
 $p_0(2) = p_{true}(2) * 0.8$
 $p_0(3) = p_{true}(3) * 0.8$
 $p_0(4) = p_{true}(4) * 1.5$
 $p_0(5) = p_{true}(5) * 2$
 $ub = 10 * p_{true}$
 $lb = 0.1 * p_{true}$
 $[p, fval, exitflag, output] = fmincon(@ohobjfun, p0, [], [], [], [], lb, ub)$
 $k_p = 1;$
 $p_p(:, k_p) = \underline{p}$
 $P_{x1} = (\frac{180}{\pi})P_{x1}$
 $P_{x2} = (\frac{180}{\pi})P_{x2}$
 $P_z = (\frac{180}{\pi})P_z$

A.1.2 Minimalisering

I minimaliseringen blir matrisene seendes slik ut :

$$F = F(p) = \begin{bmatrix} 0 & 0 \\ 0 & -\frac{1}{p(1)} \end{bmatrix} Q = Q(p) = \begin{bmatrix} p(4)^2 & 0 \\ 0 & \frac{2*p(3)^2}{p(1)} \end{bmatrix} H = \begin{bmatrix} 1 & 1 \end{bmatrix}$$

$$\hat{P} = \begin{bmatrix} p(2)^2 & 0 \\ 0 & p(3)^2 \end{bmatrix} \hat{x}_0 = \begin{bmatrix} 0 & 0 \end{bmatrix}^T$$

function J = objfun(p)

```

global N Z p_p k_p
Ts = 0.001;
p_p(:, k_p) = p; k_p = k_p + 1;
R_d = p(5)^2;
Diskretisering :
Φ = eFTs;
Γ = ohkp2dpGa(F, G, Q, Ts);
J0 = 0;
for k = 0 to N - 1
    kalmanfilter :
    TO :
    x̄ = Φx̂;
    P̄ = ΦP̂ΦT + ΓΓT;
    Kriteriet :
    S = H̄P̄HT + Rd;
    z̄ = Hx̄
    J = J - ½ln||S|| - ½(zk - z̄)TS-1(zk - z̄)
    MO :
    K = P̄HT(HP̄HT + Rd)-1;
    P̂ = (I - KH)P̄;
    x̂ = x̄ + K(zk - Hx̄);
end
J = -JN
end

```

IMatlab :

$$\hat{P} = Pe;$$

$$\Phi = Fi;$$

$$\Gamma = Ga;$$

$$\bar{x} = xp;$$

$$\hat{x} = xe;$$

$$\underline{z}_k = Z(:, k);$$

$$\hat{P} = Pe$$

$$\bar{z} = zp;$$

$$S = S;$$

Tillegg B

Datablad

Vedlagt ligger et utdrag fra databladet til hver av gyroskopene SAR100 og STIM202.

HIGH PERFORMANCE GYRO



SAR150

PRODUCT BRIEF

FEATURES

- Up to $\pm 1500^\circ/\text{s}$ input range
- Horizontal or vertical mounting
- $50^\circ/\text{h}$ in-run bias stability
- $0.65^\circ/\sqrt{\text{h}}$ angular random walk
- $0.03^\circ/\text{s/g}$ linear acceleration effect
- 5000g shock capability
- Sensor and ASIC automotive qualified according to AEC-Q100



DESCRIPTION

The SAR150 gyro contains a ButterflyGyro™ MEMS die and a BiCMOS mixed mode ASIC, housed in a rigid custom ceramic LCC package to accommodate both horizontal and vertical mounting. The sensor is factory-calibrated and compensated for temperature effects to provide high-accuracy digital output over a broad temperature range.

Input ranges

SAR150 angular rate sensor is offered with standard full-scale input ranges of $400^\circ/\text{s}$, $250^\circ/\text{s}$ or $100^\circ/\text{s}$. For custom requirements, it can also be offered with non-standard ranges up to $1500^\circ/\text{s}$.

Reliability and robustness

By utilizing a unique sealed cavity technology, the vibrating masses of the sensor are contained within the low-pressure hermetic environment needed for creating low dynamic damping and high Q factors, without any degradation over lifetime of the device whatever the

environmental conditions.

A closed-loop force feedback operation with electrostatic tuning of excitation and detection frequencies, as well as perfect mechanical and electrical balancing of the vibrating masses, results in exceptionally low shock and vibration sensitivity.

Interface

An SPI interface of the device enables easy and effective communication to the application. Serial data bus frequencies up to 8.5MHz are supported, and the angular rate data output format is 12-bit 2's complement at a maximum sampling rate of 2000SPS.

Self diagnostics

A number of functions are available through the digital SPI interface of the device, including advanced self diagnostics.

A status register contains rate overflow information (recoverable error), excitation

and detection loop diagnosis bits, a supply voltage information bit etc. In case of any irregularities detected, a specific error code is reported. This error code cannot be misinterpreted as valid sensor data.

Applications

SAR150 is a product with high bias and scale factor stability, designed for demanding applications. It is highly rugged, consumes only 50mW and offers the best value of any gyro product in its class of performance.

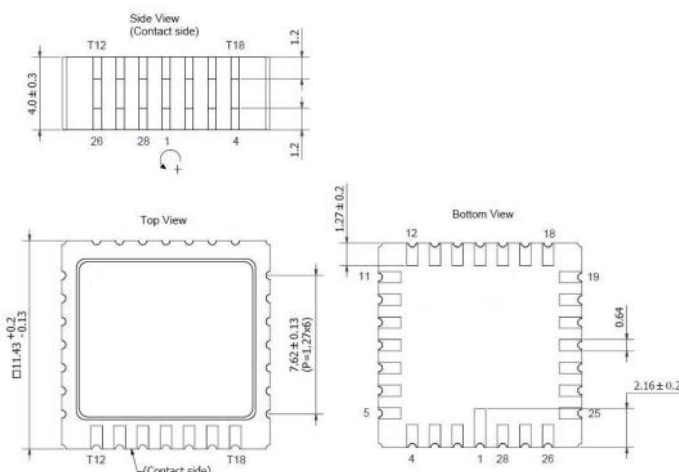
The device is ideal for platform stabilization (e.g. antennas, cameras, etc.), robot head positioning, borehole surveys, inertial navigation systems (INSs), smart munitions, train applications, aircraft instruments, unmanned aerial vehicles (UAVs), autonomous underwater vehicles (AUVs), agricultural machines, space applications, etc.

SPECIFICATIONS

Parameter	Min	Nom	Max	Unit
Weight		1.5		g
Functional characteristics				
Input range (Ω FSI)		± 400		$^{\circ}/s$
		± 250		$^{\circ}/s$
		± 100		$^{\circ}/s$
Operating temperature	-40		90	$^{\circ}C$
Power supply	4.45	5.0	5.5	V
Supply current			17	mA
Start-up time		90		ms
Maximum sampling frequency		2000		SPS
Storage temperature	-55		125	$^{\circ}C$
Dynamic overload		5000		$^{\circ}/s$
Mechanical shock			5000	g
Angular rate measurements				
Word length		12		bit
Scale factor	$\pm 400^{\circ}/s$ (Ω FSI)	0.25		$^{\circ}/s/LSB$
	$\pm 250^{\circ}/s$ (Ω FSI)	0.25		$^{\circ}/s/LSB$
	$\pm 100^{\circ}/s$ (Ω FSI)	0.10		$^{\circ}/s/LSB$
Scale factor temperature accuracy		± 0.5		%
Bias temperature accuracy		± 0.5		$^{\circ}/s$
Sensitivity accuracy		± 1		%
In-run bias stability	$\pm 400^{\circ}/s$ (Ω FSI)	0.80		$^{\circ}/vh$
	$\pm 250^{\circ}/s$ (Ω FSI)	0.75		$^{\circ}/vh$
	$\pm 100^{\circ}/s$ (Ω FSI)	0.65		$^{\circ}/vh$
Bandwidth (-3dB)		50		Hz
Non-linearity		0.1		%FS
Linear acceleration effect			0.01	$^{\circ}/s/g$
Temperature measurements				
Word length		8		bit
Scale factor		1		$^{\circ}C/LSB$
Accuracy		± 2		$^{\circ}C$
SPI interface				
Datarate			8.5	Mbit/s

MECHANICAL DIMENSIONS

All dimensions in mm. The sensitive axis is indicated with arrow and sign.

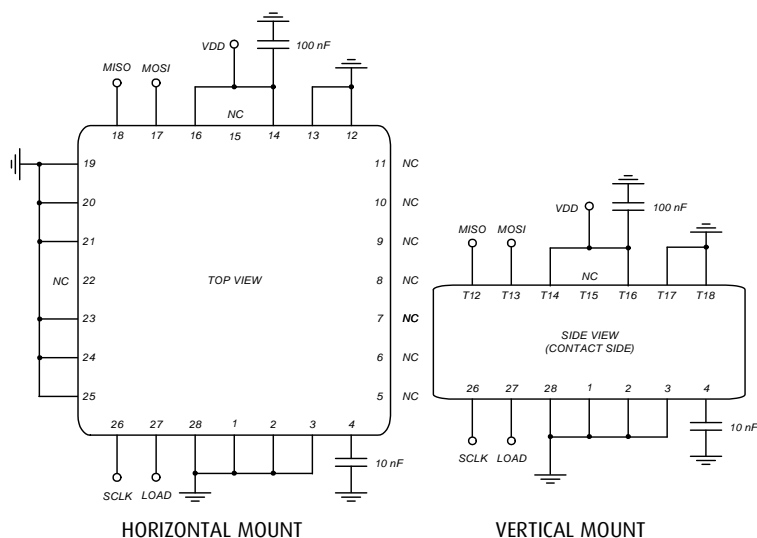


PIN OUT

Horizontal mount	Vertical mount	Pin name	Comment
1	1	GND	Connect to ground
2	2	AVSS	Connect to ground
3	3	-	Connect to ground
4	4	REFV	Connect a 10 nF low ESR de-coupling capacitor between this pin and ground
5	-	-	NC
6	-	-	NC
7	-	-	NC
8	-	-	NC
9	-	-	NC
10	-	-	NC
11	-	-	NC
12	T18	-	Connect to ground
13	T17	-	Connect to ground
14	T16	AVDD	Connect to +5 VDC ¹⁾
15	T15	-	NC
16	T14	VDD	Connect to +5 VDC ¹⁾
17	T13	MOSI	SPI data input
18	T12	MISO	SPI data output
19	-	-	Connect to ground
20	-	-	Connect to ground
21	-	-	Connect to ground
22	-	-	NC
23	-	-	Connect to ground
24	-	-	Connect to ground
25	-	-	Connect to ground
26	26	SCLK	SPI clock
27	27	LOAD	SPI load
28	28	VSS	Connect to ground

¹⁾ Connect these pins together as close as possible to the package, and from there to ground via a 100nF low ESR decoupling capacitor.

ELECTRICAL CONNECTIONS



ULTRA-HIGH PERFORMANCE GYRO MODULE



STIM202

PRODUCT BRIEF

FEATURES

- $\pm 400^\circ/\text{s}$ input range
- 1, 2 or 3 axis capability
- Electronic axis alignment
- $0.5^\circ/\text{h}$ in-run bias stability
- $0.2^\circ/\sqrt{\text{h}}$ angular random walk
- $18^\circ/\text{h/g}$ linear acceleration effect
- 1500g shock capability
- 55g module weight



DESCRIPTION

STIM is a cluster of 1, 2 or 3 high accurate MEMS-based gyros housed in a miniature package. Each axis is factory calibrated for bias and sensitivity, and compensated for temperature effects to provide high-accuracy measurements.

For many applications the excellent performance of STIM202 will replace FOG's and improve system solution with respect to robustness, reliability, size/weight, power and cost. This is accomplished by combining the well proven Sensoror ButterflyGyro™ with full digital operation.

Input range and output formats

The STIM202 full-scale angular rate input range is $400^\circ/\text{s}$. Selectable output formats are angular rate or increment angle.

Reliability and robustness

Perfect tuning of excitation and detection frequencies, as well as perfectly balanced vibrational masses, result in very low sensitivity to vibration and shock. For use

in extreme environments, the STIM202 provides a vibration isolated internal assembly to avoid rectification errors.

Interface

The unit is powered by a single +5V supply and communicates via a Plug & Play high-level RS422 interface.

Flexibility and self diagnostics

The use of a 32-bit RISC ARM microcontroller provides flexibility in configuration, e.g. for choices of output unit, sampling frequency, LP filter cut-off frequency, RS422 bit rate and line termination ON/OFF. The gyro module provides a diagnostic function, that will flag any type of errors in the system. For more advanced users, the STIM202 may be set in Service Mode, where all configuration parameters may be intermediately or permanently changed by overwriting current settings in the flash memory.

Service Mode also provides ability to perform single measurements and to access more diagnostic information.

Evaluation kits

Evaluation kits are available, supporting initial testing and device configuration. The starter kits are small, flexible and communicate via USB.

Additional PCI cards are offered for more extensive testing or characterization. The RS422 connection of the PCI card is directly compatible with the STIM202 communication cable included in the starter kit.

Applications

The STIM202 is designed to replace FOGs in general, and enables new areas of use.

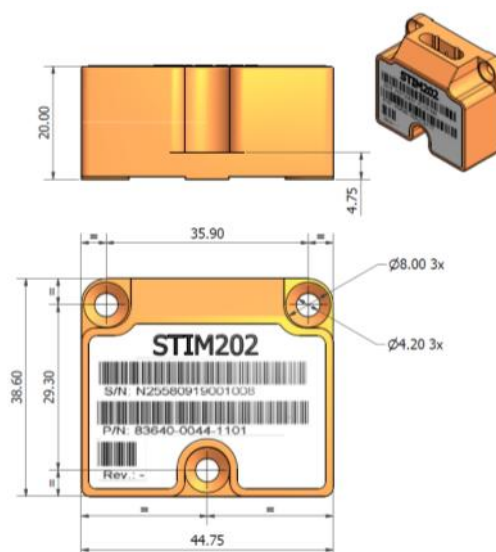
Typical applications for the product is attitude heading reference systems (AHRs), flight recorders, platform stabilization (e.g. antennas, cameras, gimbals, etc.), high performance industrial applications, unmanned aerial vehicles (UAVs), autonomous underwater vehicles (AUVs), space applications, etc.

SPECIFICATIONS

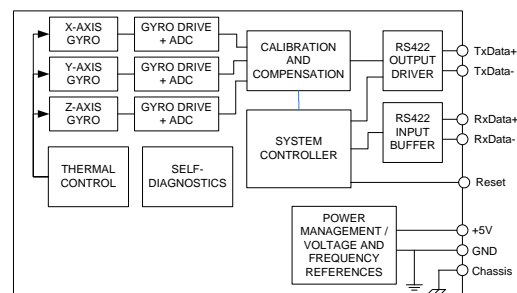
Parameter	Min	Nom	Max	Unit
Weight		55		g
Input range		±400		°/s
Resolution		24		bit
Operating temperature	-40		85	°C
Power supply	4.5	5.0	5.5	V
Supply current		200		mA
Start-up time			10	s
Sampling frequency			1000	SPS
Storage temperature	-50		90	°C
Dynamic overload			5000	°/s
Mechanical shock			1500	g
In-run bias stability		0.5		°/h
Angular random walk		0.2		°/√h
Bandwidth (-3dB)			262	Hz
Non-linearity (BSL over +/- 200 °/s)			200	ppm
Scale Factor accuracy		±0.2		%
Bias temperature accuracy (1σ)		±30		°/h rms
Linear acceleration effect			18	°/h/g
RS422 bit rate			921600	bit/s
Input resistance (termination ON)		120		Ω
Input resistance (termination OFF)		125		Ω
RESET PIN (NRST)				
Logic levels	CMOS and TTL compatible			
Minimum hold time for reset	5			ms

MECHANICAL DIMENSIONS

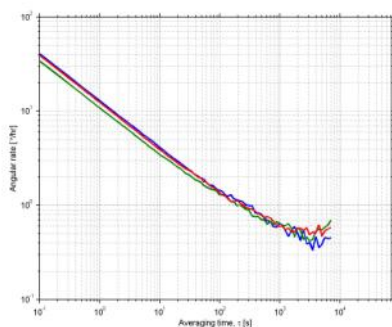
All dimensions in mm.



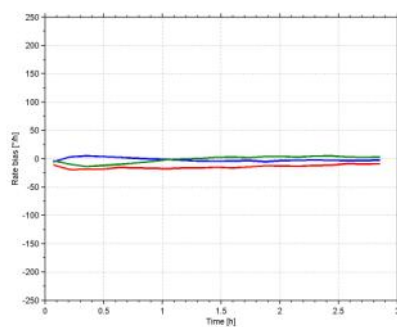
FUNCTIONAL BLOCK DIAGRAM



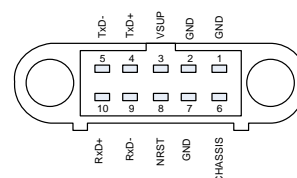
ALLAN VARIANCE



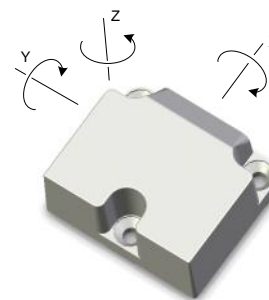
BIAS STABILITY



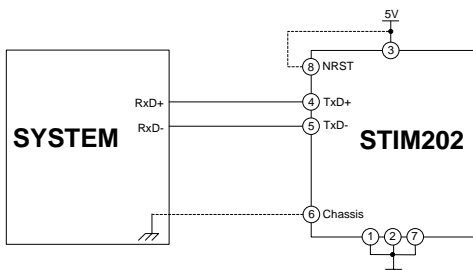
PIN OUT



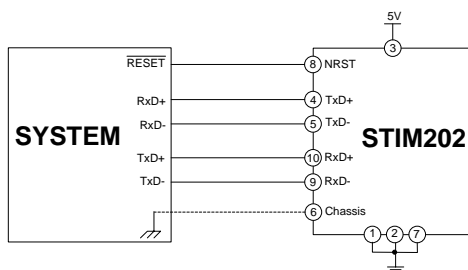
AXIS DEFINITIONS



ELECTRICAL CONNECTIONS



TRANSMIT ONLY



FULL FUNCTION

Tillegg C

MATLAB-kode

C.1 Simuleringsprogram

Simuleringsprogram med tilhørende funksjonsfiler som diskretiserer systemmatrisene.

simuleringsprogram.m

```
clear all; clc; clear figure;
%% System

T=0.02; % tidskonstant
Ts=0.001; % samplingstid
F = [0 0;0 -1/T];
G = [1 0;0 1];
H=[1 1];

%% Initialverider
cgr=4.5*(pi/180);
Pe0 = [1 0;0 (cgr)^2];
Q=[1e-10 0;0 1e-10];
Rd=(cgr)^2;

bx =chol(Pe0,'lower')*randn(2,1);
by =chol(Pe0,'lower')*randn(2,1);
bz =chol(Pe0,'lower')*randn(2,1);

%% Diskretiserer

Fi=expm(F*Ts);
Ga=ohkp2dpGa(F,G,Q,Ts);
```

```

%% Simulerer støymodell
N=1000;

%w_s=zeros(1,N+1);

for i=1:N;
    sekunder(1,i)=Ts*i;

    bx=Fi*bx+Ga*randn(2,1);
    by=Fi*by+Ga*randn(2,1);
    bz=Fi*bz+Ga*randn(2,1);

    zx(1,i) = H*bx + sqrt(Rd)*randn(1);
    zy(1,i) = H*by + sqrt(Rd)*randn(1);
    zz(1,i) = H*bz + sqrt(Rd)*randn(1);
end

%% Importere sann gyrodata
a=importdata('test2.txt');
temp=a.data;
clear a;
temp2=temp(1:1:length(zx)-1,:); % tilpasse lengden på reell data
    = simulert data
%% Plot

figure(1)
subplot(2,1,1); title('Simulert og reell vinkelhastighet (deg/
    sek) om x-aksen'), hold on
plot(sekunder,zx)
subplot(2,1,2);
legend('Simulert vinkelhastighet (deg/sek)')
plot(temp2(:,1),temp2(:,2),'r')
legend('Reell vinkelhastighet (deg/sek)')

figure(2);
subplot(2,1,1); title('Simulert og reell vinkelhastighet (deg/
    sek) om y-aksen'), hold on
plot(sekunder,zy)
legend('Simulert vinkelhastighet (deg/sek)')
subplot(2,1,2);
plot(temp2(:,1),temp2(:,3),'r')
legend('Reell vinkelhastighet (deg/sek)')

figure(3);
subplot(2,1,1); title('Simulert og reell vinkelhastighet (deg/
    sek) om z-aksen'), hold on
plot(sekunder,zz)
legend('Simulert vinkelhastighet (deg/sek)')
subplot(2,1,2);

```



```
plot(temp2(:,1),temp2(:,4),'r')
legend('Reell vinkelhastighet (deg/sek)')
```

kp2dpGa.m

```
function [Ga] = kp2dpGa(F,G,Qb,d)

[S] = kp2dpS(F,G,Qb,d);
[L,D]=chol(S);
Ga=L*sqrt(D);
```

kp2dpS.m

```
function [S] = kp2dpS(F,G,Q,d)
syms tau;
S =subs( int(expm(F*(d-tau))*G*Q*(G')*expm((F')*(d-tau)),0,d));
```

C.2 Allanvarians

Hovedskript med tilhørende funksjonsfil allan.m

allansar100.m

```
clear all;

%% Importere data
a = importdata('sensor-9.txt','\t')
temp = a.data;
clear a;

%% Skille i data-sett

tau0=0.02; % sampling = 50 hz
tid = temp(:,1); % skille ut tid
y = temp(:,2)*0.1; % skille ut sampler i grader — gyro i ro

%%Trekke fra middelerdi
for i=1:length(temp)
    y2(i,1)=y(i,1)-4;
end

variance = var(y) % Sample varians

%% Plot av gyrosignal
```

```

subplot(2,1,1); title('Gyrosignal fra SAR100'), hold on
axis tight
plot(y);
subplot(2,1,2); title('Gyrosignal fra SAR100 med fratrekk av
    middelvei'), hold on
axis tight
plot(y2);

%% Input til allan.m

    data.freq = y;
    data.time = tid;
data.rate = 50;
data.units = 'grader/sek';
tau = [];
name = 'SAR100';
verbose = 2;

%[retval, s, errorb, tau] = allan_overlap(data,tau,name,verbose)
;
[retval, s, errorb, tau] = allan(data,tau,name,verbose);
%[retval, s, errorb, tau] = allan_modified(data,tau,name,verbose
);

%% Gyro-parametre

gjennomsnitt1=mean(y);
% gjennomsnitt1=mean(y2);
b = min(retval); % bias instability grader/sekund
bias_stability = b*3600 % konverterer til grader/time

for i=1:length(tau)

    if tau(i) == 1
        ARW=retval(i)*60 % grader/sqrt(s)
    end
end

```

allanstim202.m

```
clear all;

%% Importere data
a = importdata('x-akse.txt','\t')
temp = a.data;
clear a;

%% Skille i data-sett
tau0=0.001; % sampling = 50 hz
tid = temp(:,1); % skille ut tid
y = temp(:,2); % skille ut sampler i grader — gyro i ro

variance = var(y) % Sample variants

%% Plot av gyrosignal
axis tight
plot(y);title('Gyrosignal fra STIM202'), hold on

%% Input til allan.m

data.freq = y;
data.time = tid;
data.rate = 1000;
data.units = 'grader/sek';
tau = [];
name = 'STIM202';
verbose = 2;

%[retval, s, errorb, tau] = allan_overlap(data,tau,name,verbose)
;
[retval, s, errorb, tau] = allan(data,tau,name,verbose);
%[retval, s, errorb, tau] = allan_modified(data,tau,name,verbose)
);

%% Gyro-parametre
gjennomsnitt=mean(y);
b = min(retval); % bias instability grader/sekund
bias_stability = b*3600 % konverterer til grader/time

for i=1:length(tau)

    if tau(i) == 1
        ARW=retval(i)*60 % grader/sqrt(s)
    end
end
end
```

allan.m

```

function [retval, s, errorb, tau] = allan(data,tau,name,verbose)
% ALLAN Compute the Allan deviation for a set of time-domain
% frequency data
% [RETVAL, S, ERRORB, TAU] = ALLAN(DATA,TAU,NAME,VERBOSE)
%
% Inputs:
% DATA should be a struct and have the following fields:
% DATA.freq or DATA.phase
%           A vector of fractional frequency measurements (
% df/f) in
%           DATA.freq *or* phase offset data (seconds) in
% DATA.phase .
%           If frequency data is not present, it will be
% generated by
%           differentiating the phase data.
%           If both fields are present, then DATA.freq will
% be used.
%
% DATA.rate or DATA.time
%           The sampling rate in Hertz (DATA.rate) or a
% vector of
%           timestamps for each measurement in seconds (DATA
% .time).
%           DATA.rate is used if both fields are present.
%           If DATA.rate == 0, then the timestamps are used.
%
% DATA.units (optional)
%           The units for the data. If present, the string
% DATA.units
%           is added to the plot y-axis label.
%
% TAU is an array of tau values for computing Allan deviation.
% TAU values must be divisible by 1/DATA.rate (data points
% cannot be
% grouped in fractional quantities!) and invalid values are
% ignored.
% Leave empty to use default values.
% NAME is an optional label that is added to the plot titles.
% VERBOSE sets the level of status messages:
% 0 = silent & no data plots;
% 1 = status messages & minimum plots;
% 2 = all messages and plots (default)
%
% Outputs:
% RETVAL is the array of Allan deviation values at each TAU.
% S is an optional output of other statistical measures of the
% data (mean, std, etc).

```

```

% ERRORB is an optional output containing the error estimates
%   for a 1-sigma
%   confidence interval. These values are shown on the figure
%   for each point.
% TAU is an optional output containing the array of tau values
%   used in the
% calculation (which may be a truncated subset of the input or
%   default values).
%
% Example:
%
% To compute the Allan deviation for the data in the variable "
%   lt":
% >> lt
% lt =
%     freq: [1x86400 double]
%     rate: 0.5
%
% Use:
%
% >> ad = allan(lt,[2 10 100],'lt data',1);
%
% The Allan deviation will be computed and plotted at tau =
%   2,10,100 seconds.
% 1-sigma confidence intervals will be indicated by vertical
%   lines at each point.
% You can also use the default settings, which are usually a
%   good starting point:
%
% >> ad = allan(lt);
%
% Notes:
% This function calculates the standard Allan deviation (ADEV),
%   *not* the
%   overlapping ADEV. Use "allan_overlap.m" for overlapping ADEV
%   .
% The calculation is performed using fractional frequency data.
%   If only
%   phase data is provided, frequency data is generated by
%   differentiating
%   the phase data.
% No pre-processing of the data is performed, except to remove
%   any
%   initial offset (i.e., starting gap) in the time record.
% For rate-based data, ADEV is computed only for tau values
%   greater than the
%   minimum time between samples and less than the half the
%   total time. For

```

```

%   time-stamped data, only tau values greater than the maximum
%   gap between
%   samples and less than half the total time are used.
%   The calculation for fixed sample rate data is *much* faster
%   than for
%   time-stamp data. You may wish to run the rate-based
%   calculation first,
%   then compare with time-stamp-based. Often the differences
%   are insignificant.
%   To show the "tau bins" (y_k samples) on the data plot, set
%   the variable
%   TAUBIN to 1 (search for "#TAUBIN").
%   You can choose between loglog and semilog plotting of results
%   by
%   commenting in/out the appropriate line. Search for "#PLOTLOG
%   ".
%   I recommend installing "dsplot.m", which improves the
%   performance of
%   plotting large data sets. Download from File Exchange, File
%   ID: #15850.
%   allan.m will use dsplot.m if it is present on your MATLAB
%   path.
%   This function has been validated using the test data from NBS
%   Monograph
%   140, the 1000-point test data set given by Riley [1], and
%   the example data
%   given in IEEE standard 1139-1999, Annex C.
%   The author welcomes other validation results, see contact
%   info below.
%
%   For more information, see:
%   [1] W. J. Riley, "The Calculation of Time Domain Frequency
%       Stability,"
%   Available on the web:
%   http://www.ieee-uffc.org/frequency\_control/teaching.asp?name=
%       paper1ht
%
%
%   M.A. Hopcroft
%       mhopeng at gmail dot com
%
%   I welcome your comments and feedback!
%
%   MH Oct2010
%   v2.22 tau truncation to integer groups; tau sort
%       plotting bugfix
%   v2.20 synchronize updates across allan, allan_overlap,
%       allan_modified
%   v2.16 add TAU as output, fixed unusual error with dsplot v1.1

```

```
% v2.14 update plotting behaviour, default tau values
%
versionstr = 'allan v2.22';

% MH Jun2010
% v2.12 bugfix for rate data row/col orientation
%       add DATA.units for plotting
%       use dsplot.m for plotting
%
% MH MAR2010
% v2.1  minor interface and bugfixes
%       update data consistency check
%
% MH FEB2010
% v2.0  Consistent code behaviour for all "allan_x.m" functions:
%       accept phase data
%       verbose levels
%
% MH JAN2010
% v1.84  code cleanup
% v1.82  typos in comments and code cleanup
%       tau bin plotting changed for performance improvement
% v1.8   Performance improvements:
%       vectorize code for rate data
%       logical indexing for irregular rate data
% MH APR2008
% v1.62  loglog plot option
% v1.61  improve error handling, plotting
%       fix bug in regular data calc for high-rate data
%       fix bug in timestamp data calc for large starting gap
%       (thanks to C. B. Ruiz for identifying these bugs)
%       uses timestamps for DATA.rate=0
%       progress indicator for large timestamp data processing
% MH JUN2007
% v1.54  Improve data plotting and optional bin plotting
% MH FEB2007
% v1.5   use difference from median for plotting
%       added MAD calculation for outlier detection
% MH JAN2007
% v1.48  plotting typos fixes
% MH DEC2006
% v1.46  hack to plot error bars
% v1.44  further validation (Riley 1000-pt)
%       plot mean and std
% MH NOV2006
% v1.42  typo fix comments
% v1.4   fix irregular rate algorithm
```

```

%      irregular algorithm rejects tau less than max gap in
%      time data
%      validate both algorithms using test data from NBS
%      Monograph 140
% v1.3  fix time calc if data.time not present
%      add error bars (not possible due to bug in MATLAB R14SP3
%      )
%      remove offset calculation
% v1.24 improve feedback
% MH SEP2006
% v1.22 updated comments
% v1.2  errors and warnings
% v1.1  handle irregular interval data

% defaults
if nargin < 4, verbose=2; end
if nargin < 3, name=''; end
if nargin < 2 || isempty(tau), tau=2.^(-10:10); end

% plot "tau bins"? #TAUBIN
TAUBIN = 0; % set 0 or 1 % WARNING: this has a significant
%      impact on performance

% Formatting for plots
FontName = 'Arial';
FontSize = 14;
plotlinewidth=2;

if verbose ≥ 1, fprintf(1,'allan: %s\n\n',versionstr); end

%% Data consistency checks
if ~(isfield(data,'phase') || isfield(data,'freq'))
    error('Either ''phase'' or ''freq'' must be present in DATA.
        See help file for details. [con0]');
end
if isfield(data,'time')
    if isfield(data,'phase') && (length(data.phase) ≠ length(
        data.time))
        if isfield(data,'freq') && (length(data.freq) ≠ length(
            data.time))
            error('The time and freq vectors are not the same
                length. See help for details. [con2]');
        else
            error('The time and phase vectors are not the same
                length. See help for details. [con1]');
        end
    end
end
if isfield(data,'phase') && (any(isnan(data.phase)) || any(
    isinf(data.phase)))

```



```

        error('The phase vector contains invalid elements (NaN/
            Inf). [con3]');
    end
    if isfield(data, 'freq') && (any(isnan(data.freq)) || any(
        isinf(data.freq)))
        error('The freq vector contains invalid elements (NaN/
            Inf). [con4]');
    end
    if isfield(data, 'time') && (any(isnan(data.time)) || any(
        isinf(data.time)))
        error('The time vector contains invalid elements (NaN/
            Inf). [con5]');
    end
end
end

% sort tau vector
tau=sort(tau);

%% Basic statistical tests on the data set
if ~isfield(data, 'freq')
    if isfield(data, 'rate') && data.rate ≠ 0
        data.freq=diff(data.phase).*data.rate;
    elseif isfield(data, 'time')
        data.freq=diff(data.phase).*diff(data.time);
    end
    if verbose ≥ 1, fprintf(1, 'allan: Fractional frequency data
        generated from phase data (M=%g).\n', length(data.freq));
    end
    data.time(1)=[]; % make time stamps correspond to freq data
end
if size(data.freq,2) > size(data.freq,1), data.freq=data.freq';
end % ensure columns

s.numpoints=length(data.freq);
s.max=max(data.freq);
s.min=min(data.freq);
s.mean=mean(data.freq);
s.median=median(data.freq);
if isfield(data, 'time')
    if size(data.time,2) > size(data.time,1), data.time=data.
        time'; end % ensure columns
    s.linear=polyfit(data.time(1:length(data.freq)), data.freq,1)
    ;
elseif isfield(data, 'rate') && data.rate ≠ 0;
    s.linear=polyfit((1/data.rate:1/data.rate:length(data.freq)/
        data.rate)', data.freq,1);
else

```

```

        error('Either "time" or "rate" must be present in DATA. Type
              "help allan" for details. [err1]');
    end
    s.std=std(data.freq);

    if verbose ≥ 2
        fprintf(1, 'allan: input data statistics:\n');
        disp(s);
    end

    % center at median for plotting
    medianfreq=data.freq-s.median;
    sm=[]; sme=[];

    % Screen for outliers using 5x Median Absolute Deviation (MAD)
    criteria
    s.MAD = median(abs(medianfreq)/0.6745);
    if verbose ≥ 2
        fprintf(1, 'allan: 5x MAD value for outlier detection: %g\n'
            ,5*s.MAD);
    end
    if verbose ≥ 1 && any(abs(medianfreq) > 5*s.MAD)
        fprintf(1, 'allan: NOTE: There appear to be outliers in the
            frequency data. See plot.\n');
    end

    %%%
    % There are two cases, either using timestamps or fixed sample
    % rate:

    % Fixed Sample Rate Data
    % If there is a regular interval between measurements,
    % calculation is much
    % easier/faster
    if isfield(data, 'rate') && data.rate > 0 % if data rate was
    given
        if verbose ≥ 1, fprintf(1, 'allan: regular data (%g data
            points @ %g Hz)\n', length(data.freq), data.rate); end

        % string for plot title
        name=[name ' (' num2str(data.rate) ' Hz)'];

        % what is the time interval between data points?
        tmstep = 1/data.rate;

        % Is there time data? Just for curiosity/plotting, does not
        impact calculation

```

```

if isfield(data, 'time')
    % adjust time data to remove any starting gap; first
    % time step
    % should not be zero for comparison with freq data
    dtime=data.time-data.time(1)+mean(diff(data.time));
    if verbose ≥ 2
        fprintf(1, 'allan: End of timestamp data: %g sec.\n',
            dtime(end));
        if (data.rate - 1/mean(diff(dtime))) > 1e-6
            fprintf(1, 'allan: NOTE: data.rate (%f Hz) does
                not match average timestamped sample rate (%f
                Hz)\n', data.rate, 1/mean(diff(dtime)));
        end
    end
else
    % create time axis data using rate (for plotting only)
    dtime=(tmstep:tmstep:length(data.freq)*tmstep)'; %
    % column oriented
end

% check the range of tau values and truncate if necessary
% find halfway point of time record
halftime = round(tmstep*length(data.freq)/2);
% truncate tau to appropriate values
tau = tau(tau ≥ tmstep & tau ≤ halftime);
if verbose ≥ 2, fprintf(1, 'allan: allowable tau range: %g
    to %g sec. (1/rate to total_time/2)\n', tmstep, halftime);
end

% save the freq data for the loop
dfreq=data.freq;
% find the number of data points in each tau group
m = data.rate.*tau;
% only integer values allowed (no fractional groups of
% points)
%tau = tau(m-round(m)<1e-8); % numerical precision issues (
% v2.1)
tau = tau(m==round(m)); % The round() test is only correct
% for values < 2^53
%m = m(m-round(m)<1e-8); % change to round(m) for integer
% test v2.22
m = m(m==round(m));
%m=round(m);

if verbose ≥ 1, fprintf(1, 'allan: calculating Allan
    deviation...\n    '); end

% calculate the Allan deviation for each value of tau
k=0; tic;

```

```

for i = tau
    if verbose ≥ 2, fprintf(1, '%g ', i); end
    k=k+1;

    % truncate frequency set to an even multiple of this tau
    value
    freq=dfreq(1:end-rem(length(dfreq),m(k)));
    % group the data into tau-length groups or bins
    f = reshape(freq,m(k),[]); % Vectorize!
    % find average in each "tau group", y_k (each column of
    f)
    fa=mean(f,1);
    % first finite difference
    fd=diff(fa);
    % calculate two-sample variance for this tau
    M=length(fa);
    sm(k)=sqrt(0.5/(M-1)*(sum(fd.^2)));

    % estimate error bars
    sme(k)=sm(k)/sqrt(M+1);

    if TAUBIN == 1
        % save the binning points for plotting
        fs(k,1:length(freq)/m(k))=m(k):m(k):length(freq);
        fval{k}=mean(f,1);
    end

end % repeat for each value of tau

if verbose ≥ 2, fprintf(1, '\n'); end
calctime=toc; if verbose ≥ 2, fprintf(1, 'allan: Elapsed time
    for calculation: %e seconds\n', calctime); end

%% Irregular data (timestamp)
elseif isfield(data, 'time')
    % the interval between measurements is irregular
    % so we must group the data by time
    if verbose ≥ 1, fprintf(1, 'allan: irregular rate data (no
        fixed sample rate)\n'); end

    % string for plot title
    name=[name ' (timestamp)'];

    % adjust time to remove any initial offset or zero
    dtime=data.time-data.time(1)+mean(diff(data.time));
    %dtime=data.time;
    % where is the maximum gap in time record?

```

```

gap_pos=find(diff(dtime)==max(diff(dtime)));
% what is average data spacing?
avg_gap = mean(diff(dtime));

if verbose ≥ 2
    fprintf(1, 'allan: WARNING: irregular timestamp data (no
        fixed sample rate).\n');
    fprintf(1, '        Calculation time may be long and the
        results subject to interpretation.\n');
    fprintf(1, '        You are advised to estimate using an
        average sample rate (%g Hz) instead of timestamps.\n'
        ,1/avg_gap);
    fprintf(1, '        Continue at your own risk! (press any
        key to continue)\n');
    pause;
end

if verbose ≥ 1
    fprintf(1, 'allan: End of timestamp data: %g sec\n',
        dtime(end));
    fprintf(1, '        Average rate: %g Hz (%g sec/
        measurement)\n',1/avg_gap,avg_gap);
    if max(diff(dtime)) ≠ 1/mean(diff(dtime))
        fprintf(1, '        Max. gap: %g sec at position %d\n'
            ',max(diff(dtime)),gap_pos(1));
    end
    if max(diff(dtime)) > 5*avg_gap
        fprintf(1, '        WARNING: Max. gap in time record
            is suspiciously large (>5x the average interval)
            .\n');
    end
end

% find halfway point
halftime = fix(dtime(end)/2);
% truncate tau to appropriate values
tau = tau(tau ≥ max(diff(dtime)) & tau ≤ halftime);
if isempty(tau)
    error('allan: ERROR: no appropriate tau values (> %g s,
        < %g s)\n',max(diff(dtime)),halftime);
end

% save the freq data for the loop
dfreq=data.freq;
dtime=dtime(1:length(dfreq));

if verbose ≥ 1, fprintf(1,'allan: calculating Allan
    deviation...\n'); end

```

```

k=0; tic;
for i = tau
    if verbose ≥ 2, fprintf(1, '%d ', i); end

    k=k+1; fa=[]; %f=[];
    km=0;

    % truncate data set to an even multiple of this tau
    value
    freq=dfreq(dtime ≤ dtime(end)-rem(dtime(end), i));
    time=dtime(dtime ≤ dtime(end)-rem(dtime(end), i));
    %freq=dfreq;
    %time=dtime;

    % break up the data into groups of tau length in sec
    while i*km < time(end)
        km=km+1;

        % progress bar
        if verbose ≥ 2
            if rem(km, 100)==0, fprintf(1, '.'); end
            if rem(km, 1000)==0, fprintf(1, '%g/%g\n', km, round
                (time(end)/i)); end
        end

        f = freq(i*(km-1) < time & time ≤ i*km);
        f = f(~isnan(f)); % make sure values are valid

        if ~isempty(f)
            fa(km)=mean(f);
        else
            fa(km)=0;
        end

        if TAUBIN == 1 % WARNING: this has a significant
            impact on performance
            % save the binning points for plotting
            %if find(time ≤ i*km) > 0
                fs(k, km)=max(time(time ≤ i*km));
            %else
                if isempty(fs(k, km))
                    fs(k, km)=0;
                end
            end
            fval{k}=fa;
        end % save tau bin plot points

    end
end

```

```

        if verbose ≥ 2, fprintf(1, '\n'); end

        % first finite difference of the averaged results
        fd=diff(fa);
        % calculate Allan deviation for this tau
        M=length(fa);
        sm(k)=sqrt(0.5/(M-1)*(sum(fd.^2)));

        % estimate error bars
        sme(k)=sm(k)/sqrt(M+1);

    end

    if verbose == 2, fprintf(1, '\n'); end
    calctime=toc; if verbose ≥ 2, fprintf(1, 'allan: Elapsed time
        for calculation: %e seconds\n', calctime); end

else
    error('allan: WARNING: no DATA.rate or DATA.time! Type "help
        allan" for more information. [err2]');
end

#####
%% Plotting

if verbose ≥ 2 % show all data

    % plot the frequency data, centered on median
    if size(dtime,2) > size(dtime,1), dtime=dtime'; end % this
        should not be necessary, but dsplot 1.1 is a little bit
        brittle
    try
        % dsplot makes a new figure
        hd=dsplot(dtime, medianfreq);
    catch ME
        figure;
        hd=plot(dtime, medianfreq);
        if verbose ≥ 1, fprintf(1, 'allan: Note: Install dsplot.m
            for improved plotting of large data sets (File
            Exchange File ID: #15850).\n'); end
        if verbose ≥ 2, fprintf(1, '                (Message: %s)\n'
            , ME.message); end
    end
    set(hd, 'Marker', '.', 'LineStyle', 'none', 'Color', 'b'); %
        equivalent to '.-'
    hold on;

```

```

% show center (0)
plot(xlim,[0 0],':k');
% show 5x Median Absolute Deviation (MAD) values
hm=plot(xlim,[5*s.MAD 5*s.MAD],'-r');
plot(xlim,[-5*s.MAD -5*s.MAD],'-r');
% show linear fit line
hf=plot(xlim,polyval(s.linear,xlim)-s.median,'-g');
title(['Data: ' name], 'FontSize',FontSize+2, 'FontName',
    FontName);
%set(get(gca,'Title'),'Interpreter','none');
xlabel('Time [sec]', 'FontSize',FontSize, 'FontName',FontName)
;
if isfield(data,'units')
    ylabel(['data - median(data) [' data.units ']' ], '
        FontSize',FontSize, 'FontName',FontName);
else
    ylabel('freq - median(freq)', 'FontSize',FontSize, '
        FontName',FontName);
end
set(gca,'FontSize',FontSize, 'FontName',FontName);
legend([hd hm hf],{'data (centered on median)', '5x MAD
    outliers', ['Linear Fit (' num2str(s.linear(1), '%g') ') '
    ]}, 'FontSize',max(10,FontSize-2));
% tighten up
xlim([dtime(1) dtime(end)]);

% Optional tau bin (y_k samples) plot
if TAUBIN == 1
    % plot the tau divisions on the data plot
    rfs=size(fs,1);
    colororder=get(gca,'ColorOrder');
    axis tight; kc=2;
    %ap=axis;
    for j=1:rfs
        kc=kc+1; if rem(kc,length(colororder))==1, kc=2; end
        %for b=1:max(find(fs(j,:))); % new form of "find" in
        r2009a
        for b=1:find(fs(j,:), 1, 'last' );
            % plot the tau division boundaries
            %plot([fs(j,b) fs(j,b)], [ap(3)*1.1 ap(4)
                *1.1], '-', 'Color', colororder(kc,:));
            % plot tau group y values
            if b == 1
                plot([dtime(1) fs(j,b)], [fval{j}(b)-s.median
                    fval{j}(b)-s.median], '-', 'Color',
                    colororder(kc,:), 'LineWidth',4);
            else

```



```

        plot([fs(j,b-1) fs(j,b)], [fval{j}(b)-s.
            median fval{j}(b)-s.median], '-', 'Color',
            colororder(kc,:), 'LineWidth', 4);
    end
end
end
axis auto
end % End optional bin plot

end % end plot raw data

if verbose ≥ 1 % show ADEV results

    % plot Allan deviation results
    if ~isempty(sm)
        figure

        % Choose loglog or semilogx plot here      #PLOTLOG
        %semilogx(tau, sm, '-b', 'LineWidth', plotlinewidth, '
            MarkerSize', 24);
        loglog(tau, sm, '-b', 'LineWidth', plotlinewidth, '
            MarkerSize', 24);

        % in R14SP3, there is a bug that screws up the error
        % bars on a semilog plot.
        % When this is fixed in a future release, uncomment
        % below to use normal errorbars
        %errorbar(tau, sm, sme, '-b'); set(gca, 'XScale', 'log');
        % this is a hack to approximate the error bars
        hold on; plot([tau; tau], [sm+sme; sm-sme], '-k', '
            LineWidth', max(plotlinewidth-1, 2));

        grid on;
        title(['Allan Deviation: ' name], 'FontSize', FontSize+2, '
            FontName', FontName);
        %set(get(gca, 'Title'), 'Interpreter', 'none');
        xlabel('\tau [sekunder]', 'FontSize', FontSize, 'FontName',
            FontName);
        if isfield(data, 'units')
            ylabel(['\sigma_y(\tau) [' data.units ']'], 'FontSize
                ', FontSize, 'FontName', FontName);
        else
            ylabel('\sigma_y(\tau)', 'FontSize', FontSize, '
                FontName', FontName);
        end
        set(gca, 'FontSize', FontSize, 'FontName', FontName);
        % expand the x axis a little bit so that the errors bars
        look nice
    end
end

```

```
adax = axis;  
axis([adax(1)*0.9 adax(2)*1.1 adax(3) adax(4)]);  
  
% display the minimum value  
fprintf(1, 'allan: Minimum ADEV value: %g at tau = %g  
seconds\n', min(sm), tau(sm==min(sm)));  
  
elseif verbose ≥ 1  
    fprintf(1, 'allan: WARNING: no values calculated.\n');  
    fprintf(1, '        Check that TAU > 1/DATA.rate and TAU  
        values are divisible by 1/DATA.rate\n');  
    fprintf(1, 'Type "help allan" for more information.\n\n');  
    ;  
end  
  
end % end plot ADEV data  
  
retval = sm;  
errorb = sme;  
  
return
```

C.3 Maximum likelihood metoden

Hovedskript med tilhørende funksjonsfil stim202funk.m

stim202.m

```
clear all;
global N Z PA_p k_p
%% Initialisering
N=100000;
Z = zeros(1,N);
PA_t=zeros(1,N);

%% Importere data
a=importdata('y-akse.txt');
temp=a.data;
clear a;
Z=temp(:,3);

%% Kall av objfun
% Initialverdier
p0(1) = 0.5;
p0(2) = 0.5;
p0(3) = 1;
% Definere grenser
ptrue=[0.2 0.1 0.5];

p0

k_p=1;

%options=optimset('LargeScale','off');
%[p,fval,exitflag,output]=fminunc(@objfun,p0,options)
%[p,fval,exitflag,output] = fmincon(fun,x0,A,b,Aeq,beq,lb,ub)

lb=0.1*ptrue
ub=10*ptrue

[p,fval,exitflag,output] = fmincon(@stim202funk,p0,[],[],[],[],
    lb,ub)

PA_p(:,k_p)=p';

%% Plot
figure(1)
    subplot(2,1,1)
        plot(PA_p(1,:)); title('T = p(1)'), hold on
```

```
subplot(2,1,2)
plot(PA_p(2:3,:)); title('P(2,2) = p(2) og Rd = p(3)'),
    hold on
legend('P(2,2)', 'Rd')
```

stim202funk.m

```

function J = stim202funk(p)

    global N Z PA_p k_p

    Ts = 0.001;
    F = [0 0;0 -1/p(1)];
    G = [1 0;0 1];
    PA_p(:,k_p)=p'; k_p=k_p+1;
    Pe = [1 0;0 p(2)^2];
    Q = [1e-10 0;0 2*p(2)^2/p(1)];
    Rd = p(3)^2;

    Fi = expm(F*Ts);
    Ga = kp2dpGa(F,G,Q,Ts);

    xe=[0;0];
    H=[1 1];
    I=eye(2);
    J = 0;
    for k=1:N
        % TO
        xp=Fi*xe;
        Pp=Fi*Pe*Fi' + Ga*Ga';

        %Kriteriet
        zp=H*xp; S=H*Pp*H' + Rd;
        J= J -0.5*(log(det(S))) -0.5*((Z(k,1)-zp)'*inv(S)*(Z(k,1)-zp));

        %MO
        K=Pp*H'*inv(S);
        Pe=(I-K*H)*Pp;
        xe=xp + K*(Z(k,1)-zp);

    end
    J=-J;
    % disp(['i=',num2str(i),'k=',num2str(k),' J=',num2str(J(:,end))])
end

```

C.4 Autokorrelasjon

Scriptet ble brukt til å plotte autokorrelasjon av signaler.

autokorrelasjon.m

```
a=importdata('sensor-9.txt');
temp=a.data;
clear a;
Ts=0.02

auto=xcorr(temp(:,2));
figure(1)

%subplot(4,1,1); title('Autokorrelasjon for hvitstøy'), hold
on
    t1=(-((length(temp(:,2))-1)):((length(temp(:,2))-1))*Ts
        ;
    plot(t1,auto)
```